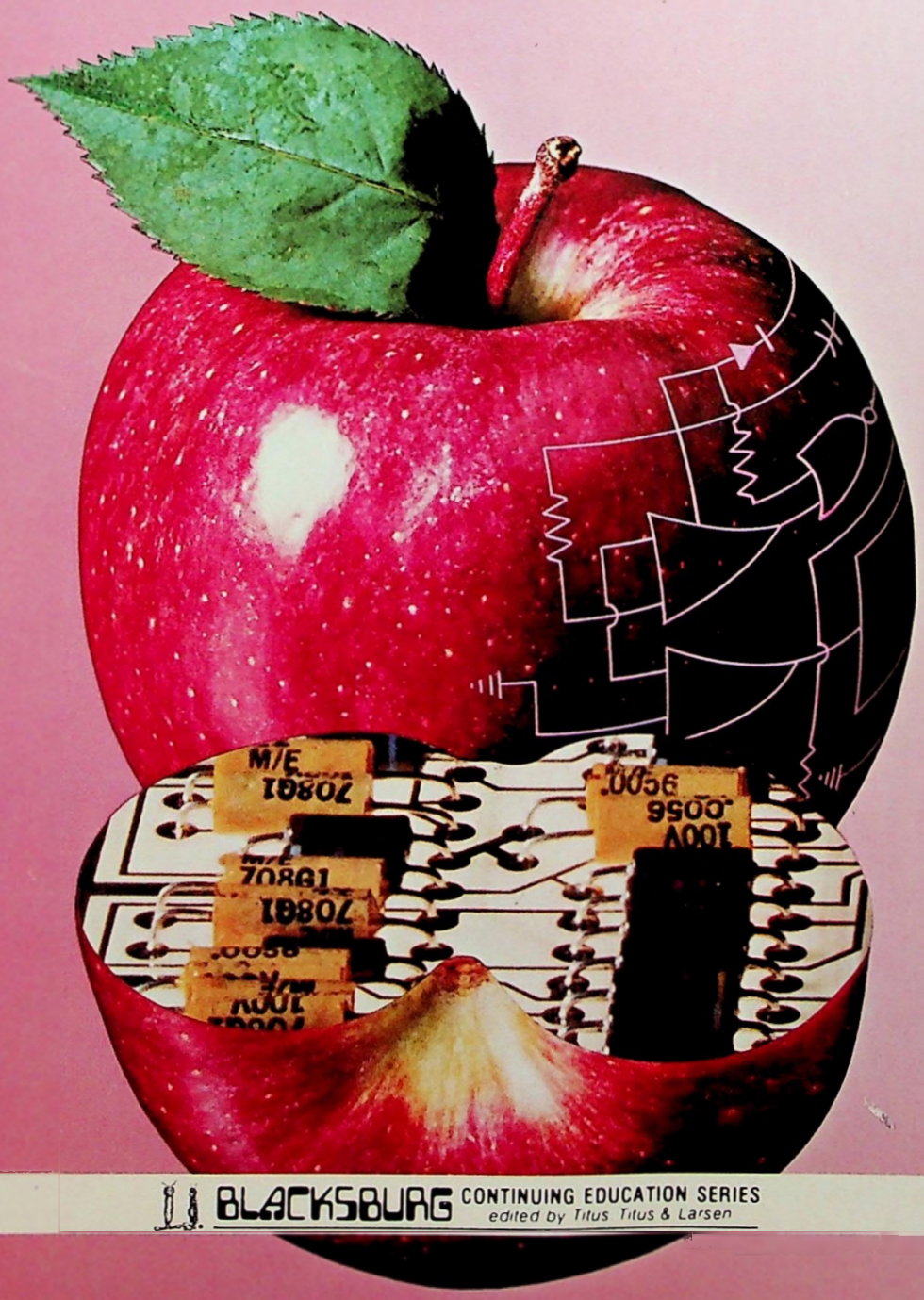
21863

CIRCUIT DESIGN PROGRAMS FOR THE APPLE II®

050A PT 151
NCE 15.95
2J 2072

BY
HOWARD M. BERLIN



BLACKSBURG CONTINUING EDUCATION SERIES
edited by Titus Titus & Larsen

The Blacksburg Continuing Education™ Series

The Blacksburg Continuing Education Series™ of books provide a Laboratory—or experiment-oriented approach to electronic topics. Present and forthcoming titles in this series include:

- Advanced 6502 Interfacing
- Analog Instrumentation Fundamentals
- Apple Interfacing
- Basic Business Software
- BASIC Programmer's Notebook
- Circuit Design Programs for the Apple II
- Circuit Design Programs for the TRS-80
- Design of Active Filters, With Experiments
- Design of Op-Amp Circuits, With Experiments
- Design of Phase-Locked Loop Circuits, With Experiments
- Design of Transistor Circuits, With Experiments
- Design of VMOS Circuits, With Experiments
- 8080/8085 Software Design (2 Volumes)
- 8085A Cookbook
- Electronic Music Circuits
- 555 Timer Applications Sourcebook, With Experiments
- Guide to CMOS Basics, Circuits, & Experiments
- How to Program and Interface the 6800
- Introduction to FORTH
- Microcomputer–Analog Converter Software and Hardware Interfacing
- Microcomputer Interfacing With the 8255 PPI Chip
- Microcomputer Design and Maintenance
- NCR Basic Electronics Course, With Experiments
- NCR Data Communications Concepts
- NCR Data Processing Concepts Course
- NCR EDP Concepts Course
- PET Interfacing
- Programming and Interfacing the 6502, With Experiments
- 16-Bit Microprocessors
- 6502 Software Design
- 6801, 68701, and 6803 Microcomputer Programming and Interfacing
- The 68000: Principles and Programming
- 6809 Microcomputer Programming & Interfacing, With Experiments
- TEA: An 8080/8085 Co-Resident Editor/Assembler
- TRS-80 Assembly Language Made Simple
- TRS-80 Interfacing (2 Volumes)
- TRS-80 More Than BASIC

In most cases, these books provide both text material and experiments, which permit one to demonstrate and explore the concepts that are covered in the book. These books remain among the very few that provide step-by-step instructions concerning how to learn basic electronic concepts, wire actual circuits, test microcomputer interfaces, and program computers based on popular microprocessor chips. We have found that the books are very useful to the electronic novice who desires to join the "electronics revolution," with minimum time and effort.

Jonathan A. Titus, Christopher A. Titus, and David G. Larsen
"The Blacksburg Group"

Bug symbol trademark Nanotran, Inc., Blacksburg, VA 24060

Circuit Design Programs for the Apple II®

by
Howard M. Berlin

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

Copyright © 1982 by Howard M. Berlin, Ltd.

FIRST EDITION
FIRST PRINTING—1982

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-21863-1
Library of Congress Catalog Card Number: 81-85516

Edited by: *C. Herbert Feltner*
Illustrated by: *T. R. Emrick*

Printed in the United States of America.

Preface

The great mathematician Gottfried Wilhelm Leibniz perhaps stated it best when he said: "It is unworthy of excellent men to lose hours like slaves in the labor of calculation." Before there were computers, virtually every aspect of electronic circuit design and analysis was performed either by hand, or with a slide rule. When done by hand, there was always a good chance that a computational error would exist. With a slide rule, the preciseness of the result left much to be desired.

With the great strides made in the integrated circuit revolution, there now exist a number of relatively inexpensive "personal-type" digital computers, such as the Apple II®, which can efficiently perform virtually any type of computation required for the design and analysis of electronic networks. Consequently, it is the goal of this book to present a variety of useful BASIC language programs which will greatly simplify the design and analysis of commonly encountered circuit problems.

All of the programs listed in this book were written for an Apple II "Plus"® or an Apple II having the Applesoft® firmware card installed. In either case, there should be 32K of random access memory although 48K is preferred. Applesoft BASIC offers many advantages over the integer BASIC set of commands and statements, particularly the mathematical functions, many of which are used repeatedly in the programs described in this book. However, many of these routines can be translated to run on other computer systems such as the PET® and TRS-80®. The program listings and the actual output results, as they would normally appear on the video display, were printed using a "daisywheel" printer instead of one using a dot-matrix font to make the reading easier.

In eight chapters, a variety of programs are presented which enable the user to solve a myriad of problems relating to plotting and the simple statistical verification of experimental data; rms and average values, as well as the Fourier series of a periodic waveform; inverse Laplace, Fourier, and other network transforms; design of matching pads, attenuators, active filters, heat sinks, integrated circuit timer, zener diode regulator, and bipolar transistor circuits; and the solution of simultaneous equations with real or complex coefficients, damped oscillations, and polynomial roots. Some of the programs, such as plot routines using HiRes graphics, are written as subroutines so that they may be used as part of some of your own programs.

All output is displayed on the video screen, but most programs can be easily modified so that numerical results are handled by a line printer. The least squares regression program has the capability to store data on disk with a given file name. In addition, each program is assigned a name, which may be used as the file specification for those of you storing programs on disk media.

Several of the programs in this book were modified from programs originally written in FORTRAN, which were part of extensive subroutine packages. Others were adapted from BASIC programs that have been published elsewhere; however, the algorithms are standard. The mathematical theory for a particular circuit's design, or numerical technique used was intentionally kept to a minimum, since this book was not meant to replace traditional texts on circuit theory or numerical analysis and methods. However, there are short descriptions of the programs, along with computed examples and applicable schematic diagrams, which are used to aid in both the explana-

tion and verification of results against your own copies of the programs.

All the programs listed in this book are available on diskettes, prepared in the 13-sector DOS 3.2 (and earlier) format. In addition, the DOS has been specially modified to allow you to BOOT any one of the seven diskettes on a DOS 3.3 system. Further information on these programs will be found at the end of this book.

HOWARD M. BERLIN

This book is dedicated to Mark Samuel

Contents

CHAPTER 1

ABOUT THE PROGRAMS 7

Basic Facts—Using a Line Printer—Graphics—Program Listings

CHAPTER 2

PLOTTING GRAPHS 9

Histograms—Cartesian Plots—Semilogarithmic Plots—Log-Log Plots—Polar Plots—Plotting with a Line Printer

CHAPTER 3

THE ANALYSIS OF SIGNALS 33

The Average and RMS Values of a Waveform—Fourier Series—Fourier Transform—Analysis of Damped Oscillations

CHAPTER 4

BASIC STATISTICS AND THE ANALYSIS OF DATA 47

Least Squares Regression Curves—Sample Statistics

CHAPTER 5

NETWORKS AND TRANSFORMS 61

Roots of Polynomials—The Inverse Laplace Transform—Impedance Matching Pads—Pi-Tee (Delta-Wye) Transforms—Mesh Current Network Analysis

CHAPTER 6

ACTIVE FILTER DESIGN 83

Low-Pass and High-Pass Filters—The State-Variable Filter—A Bandpass Filter for Q Less Than 50—Staggered-Tuned Butterworth Bandpass Filters—The Notch Filter—A Wideband Filter

CHAPTER 7

SOLID-STATE DEVICES 111

Zener Diode Voltage Regulation Design—555 Timer Design—Transistors—Heat Sink Design

CHAPTER 8

ADDITIONAL ROUTINES 123

4-Quadrant Arctangent Function—Complex Number Math—Minimum and Maximum Values of an Array

APPENDIX

STANDARD RESISTOR AND CAPACITOR VALUES 129

Resistors—Capacitors

INDEX 131

1

About the Programs

BASIC FACTS

The programs described in this book were written for either an Apple II "Plus"® system or an Apple II with the Applesoft® card installed. In either case, the system should have a minimum of 32K of read/write memory, although 48K is preferred. The reasons for this system configuration are as follows:

1. Diskette Applesoft occupies approximately 10K of memory; system overhead and video occupies 2K; while DOS (the disk operating system) occupies another 10.5K. Consequently, diskette Applesoft cannot be used in Apples with less than 24K of memory.
2. Cassette Applesoft has the same memory requirement as diskette Applesoft, except that the DOS requirement is not applicable. Thus, cassette Applesoft cannot be used in Apples with less than 16K of memory.
3. However, the HGR command, used by all the video plotting programs in this book, is not supported by either the cassette or diskette Applesoft.

Although the programs can be stored on cassette tape media, diskettes are preferred for quicker loading of the programs. The REGRESSN program of Chapter 4 was written so that the data may be saved on disk after choosing an appropriate file name.

All computations are performed using single precision arithmetic, so that most numerical results will be displayed with a maximum of nine digits. For the majority of the computations, this has been shown to be more than sufficient. In other cases, the numerical calculations are provided with enough decimal places to make the results significant. As an example, resistance values calculated in kilohms are usually displayed with two decimal places.

USING A LINE PRINTER

Just about all of the programs can be easily converted to have the output results sent to a line printer rather than the video display. Other programs, such as LINEPLOT, XLOGPLOT, and POLARPLT, are specifically designed to plot the results on a line printer. Although primarily intended to be used with parallel type printers, these programs are also easily adapted for serial printers.

GRAPHICS

All video display plotting uses high resolution, or HiRes graphics, while those programs that draw a schematic diagram use LoRes graphics. It should be remembered that page 1 of the HiRes display area begins at decimal memory address 8192. If your program, after the inclusion of any one of the graphics subroutines, becomes too large, you may overlap the HiRes display area in memory and the program may "hang." Before running such a program, type:

```
PRINT PEEK (105) + PEEK (106)*256
```

and then hit the RETURN key. The number displayed is the current LOMEM. If it is greater than 8192, you must then either delete some of the program, add a line at the beginning of your program to set LOMEM to 16384 causing the variables to be stored above the HiRes display area, or you may want to set up a chain to this program so that you use less memory. Pages 141-142 of the DOS 3.2 manual details memory allocation.

PROGRAM LISTINGS

The listings for the programs discussed in this book were printed with a "daisywheel" printer to make the reading easier when compared to a dot-matrix–style printer. There are a number of programs which require exponentiation, which will be represented either by the symbols ^ or © in the listings.

2

Plotting Graphs

Perhaps one of the best methods for describing mathematical functions or experimental results is to plot them in the form of a graph. Remember, a good picture is worth 10,000 words. This chapter illustrates how it is possible to graph functions or discrete data points on the video display in the form of histograms, Cartesian, semilogarithmic, log-log, or polar plots, all of which use HiRes graphics. Most of these programs are primarily written as subroutines so that they may be easily used in conjunction with your own special programs. In addition, three programs are included which allow plotting of various types of graphs on either a serial or parallel line printer.

HISTOGRAMS

One of the simplest types of graphs is the histogram, or bar graph, which is usually used to display frequency distributions. Fig. 2-1 shows the listing of the PHISTGM subroutine used to generate a histogram on the video display. Required input variables are: the number of data points N, the horizontal and vertical values X(I) and Y(I), respectively, for each point, and the graph title T$. Fig. 2-2 illustrates a simple driver, or main program that can be used with the PHISTGM subroutine by executing a GOSUB 5000 statement as in line 155. The subroutine assumes that each X-axis value has only one corresponding Y-axis value. Consequently, a maximum of 265 individual bars may be plotted. The DIM statement in line 105 of the main program must be included because the coordinate values are stored before they are plotted on the display.

Example 2-1

Plot a histogram of the result of measuring the values of a sample of 55, 10% resistors, all of which are supposed to have the same nominal value, 100 ohms. As a result of our sampling, we find that the values vary with several resistors having like values, as listed in Table 2-1.

Using the above data, Fig. 2-3A shows the results of executing the main program of Fig. 2-2 and the PHISTGM subroutine (Fig. 2-1) as one program. After all the 14 individual points and the graph title are entered, the display informs us as to the current ranges of the two axes. The program then asks us how we want the axes scaled, such as having the horizontal axis range from 80 to 120 while the vertical axis varies from 0 to 15. Because of the method in which the axes are scaled, it is best to have the vertical axis (i.e., the "frequency" axis) incremented by a number that is easily divisible by four or eight, since there are four major divisions, or tic marks, on this axis. However, this is not mandatory. When the scales are entered, the resultant HiRes graphics histogram is drawn on the video display, as shown in Fig. 2-3B.

Table 2-1. Data for Example 2-1

Point No.	Value	Frequency
1	90	1
2	92	1
3	93	2
4	94	4
5	95	3
6	96	4
7	98	7
8	99	8
9	100	11
10	101	4
11	103	2
12	104	3
13	106	3
14	110	2
		total = 55

```
5000  REM  HISTOGRAM PLOT SUBROUTINE FOR VIDEO DISPLAY (PHISTGM)
5001  REM  X(I)=HORIZ.AXIS
5002  REM  Y(I)=VERTICAL AXIS
5003  REM  N=NO.OF DATA PTS.
5004  REM  T$=GRAPH TITLE
5005  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
5006  FOR I = 2 TO N
5007  IF (Y1 - Y(I)) <  = 0 THEN 5010
5008  Y1 = Y(I)
5009  GOTO 5012
5010  IF (Y2 - Y(I)) >  = 0 THEN 5012
5011  Y2 = Y(I)
5012  IF (X1 - X(I)) <  = 0 THEN 5015
5013  X1 = X(I)
5014  GOTO 5017
5015  IF (X2 - X(I)) >  = 0 THEN 5017
5016  X2 = X(I)
5017  NEXT I
5018  PRINT "MIN,MAX X VALUES ARE:  ";X1;"-->";X2
5019  PRINT "MIN,MAX Y VALUES ARE:  ";Y1;"-->";Y2
5020  PRINT
5021  INPUT "MIN,MAX X-AXIS SCALE: ? ";X1,X2
5022  INPUT "MIN,MAX Y-AXIS SCALE: ? ";Y1,Y2
5023  XD = X2 - X1:YD = Y2 - Y1
5024  REM  PLOT AXES WITH SCALES
5025  IF YD = 0 THEN YD = 1
5026  IF XD = 0 THEN XD = 1
5027  HCOLOR= 7: HGR
5028  REM  VERTICAL AXIS ------
5029  HPLOT 8,0 TO 8,152
5030  FOR Y = 152 TO 0 STEP  - (152 / 8): HPLOT 5,Y TO 7,Y: NEXT Y
5031  FOR Y = 152 TO 0 STEP  - (152 / 4): HPLOT 2,Y TO 7,Y: NEXT Y
5032  REM  HORIZ AXIS ---------
5033  HPLOT 13,154 TO 278,154
5034  FOR X = 13 TO 278 STEP 265 / 8: HPLOT X,155 TO X,157: NEXT X
5035  FOR X = 13 TO 278 STEP 265 / 4: HPLOT X,157 TO X,159: NEXT X
5036  PRINT : VTAB 24: PRINT "Y-AXIS ";Y1;":";Y2; SPC( 4)"X-AXIS ";X1;":"
      ;X2: PRINT
5037  FOR I = 1 TO N
5038  X = 278 - ((X2 - X(I)) * 265 / XD)
5039  Y = 152 - ((Y(I) - Y1) * 152 / YD)
5040  IF X > 278 OR X < 13 THEN 5044
5041  YT = Y
5042  IF YT < 0 THEN YT = 0
5043  FOR Y = YT TO 154: HPLOT X,Y: NEXT Y: REM  154 FOR EVEN BOTTOM LINE
5044  NEXT I: INVERSE
5045  PRINT "-";T$;"-";: NORMAL : PRINT  SPC( 28 -  LEN (T$))"CONTINUE?";
      : INPUT "";A$
5046  PRINT : PRINT "ANY CHANGES IN SCALE FACTORS (Y/N) ?";: INPUT "";A$
5047  IF A$ <  > "Y" THEN  RETURN
5048  GOTO 5018
```

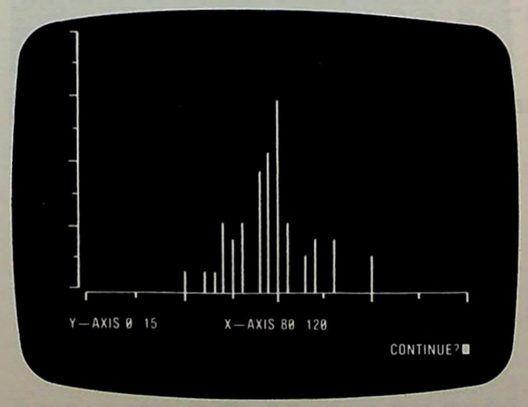
Fig. 2-1. Listing for PHISTGM subroutine.


```
100  REM  SAMPLE MAIN PROGRAM
105  DIM X(265),Y(265):SP$ = " "
110  TEXT : HOME : INPUT "NUMBER OF DATA POINTS? ";N
115  PRINT : PRINT : FOR I = 1 TO N
116  IF I > 9 THEN SP$ = ""
120  PRINT "POINT ";SP$;I;
125  INPUT "   X = ";X(I)
130  CALL  - 998
135  HTAB 26
140  INPUT "Y = ";Y(I)
145  NEXT I
150  PRINT : INPUT "TITLE? ";T$
155  PRINT : GOSUB 5000: TEXT : HOME
160  END : STOP : END : STOP
```

Fig. 2-2. Simple main program for PHISTGM.

```
NUMBER OF DATA POINTS ?  14

POINT 1    X=90        Y=1
POINT 2    X=92        Y=1
POINT 3    X=93        Y=2
POINT 4    X=94        Y=4
POINT 5    X=95        Y=3
POINT 6    X=96        Y=4
POINT 7    X=98        Y=7
POINT 8    X=99        Y=8
POINT 9    X=100       Y=11
POINT 10   X=101       Y=4
POINT 11   X=103       Y=2
POINT 12   X=104       Y=3
POINT 13   X=106       Y=3
POINT 14   X=110       Y=2

TITLE ?  100 OHM RESISTOR SAMPLE

MIN,MAX X VALUES ARE:  90-->110
MIN,MAX Y VALUES ARE:  1-->11

MIN,MAX X-AXIS SCALE:  ?  80,120
MIN,MAX Y-AXIS SCALE:  ?  0,15
```

(A) Output results.

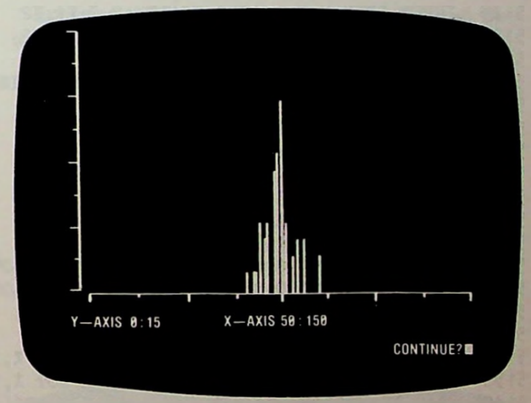


(B) Video plot.

Fig. 2-3. Plot of resistor histogram.

```
ANY CHANGE IN SCALE FACTORS (Y/N)  ?  Y

MIN,MAX X VALUES ARE:  80-->120
MIN,MAX Y VALUES ARE:  0-->15

MIN,MAX X-AXIS SCALE:  ?  50,150
MIN,MAX Y-AXIS SCALE:  ?  0,15
```

(A) Entries for new scales.



(B) Video plot.

Fig. 2-4. Histogram of resistors with new scales.

The histogram then remains on the screen until the RETURN key is pressed, after which we are asked whether or not we are satisfied with how the axes were scaled, as shown by the new entries in Fig. 2-4A and the resultant graph in Fig. 2-4B. The process of changing the axes scales can be repeated as many times as desired.

CARTESIAN PLOTS

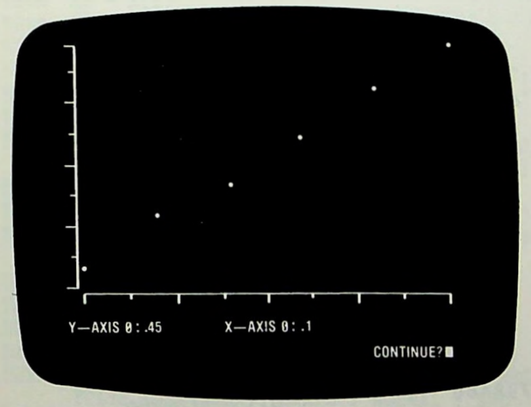
The program for generating a Cartesian, or rectangular X-Y coordinate, plot is similar to the PHISTGM subroutine. The PCARTXY subroutine is shown in Fig. 2-5, and can be used with the

```
5100  REM  CARTESIAN X-Y PLOT SUBROUTINE FOR VIDEO DISPLAY (PCARTXY)
5101  REM  HOWARD M. BERLIN LTD.
5102  REM  X(I)=HORIZ.AXIS
5103  REM  Y(I)=VERTICAL AXIS
5104  REM  N=NO.OF DATA PTS.
5105  REM  T$=GRAPH TITLE
5106  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
5107  FOR I = 2 TO N
5108  IF (Y1 - Y(I)) <  = 0 THEN 5111
5109  Y1 = Y(I)
5110  GOTO 5113
5111  IF (Y2 - Y(I)) >  = 0 THEN 5113
5112  Y2 = Y(I)
5113  IF (X1 - X(I)) <  = 0 THEN 5116
5114  X1 = X(I)
5115  GOTO 5118
5116  IF (X2 - X(I)) >  = 0 THEN 5118
5117  X2 = X(I)
5118  NEXT I
5119  PRINT
5120  PRINT "MIN,MAX X VALUES ARE:  ";X1;"-->";X2
5121  PRINT "MIN,MAX Y VALUES ARE:  ";Y1;"-->";Y2
5122  PRINT
5123  INPUT "MIN,MAX X-AXIS SCALE: ? ";X1,X2
5124  INPUT "MIN,MAX Y-AXIS SCALE: ? ";Y1,Y2
5125  XD = X2 - X1:YD = Y2 - Y1
5126  HOME : VTAB 24: INPUT "FILL BETWEEN POINTS ? ";FILL$
5127  PRINT : PRINT
5128  IF XD = 0 THEN XD = 1
5129  IF YD = 0 THEN YD = 1
5130  REM   PLOT AXES WITH SCALES
5131  HCOLOR= 7: HGR
5132  REM  VERTICAL AXIS--------
5133  HPLOT 8,1 TO 8,150
5134  FOR Y = 150 TO 0 STEP  - (150 / 8): HPLOT 5,Y TO 7,Y: NEXT Y
5135  FOR Y = 150 TO 0 STEP  - (150 / 4): HPLOT 3,Y TO 7,Y: NEXT Y
5136  REM   HORIZ AXIS ----------
5137  HPLOT 13,154 TO 279,154
5138  FOR X = 13 TO 279 STEP 266 / 8: HPLOT X,155 TO X,157: NEXT X
5139  FOR X = 13 TO 279 STEP 266 / 4: HPLOT X,157 TO X,159: NEXT X
5140  VTAB 24: PRINT "Y-AXIS ";Y1;":";Y2; SPC( 4)"X-AXIS ";X1;":";X2: PRINT
5141  FOR I = 1 TO N
5142  X = 279 - ((X2 - X(I)) * 266 / XD)
5143  Y = 150 - ((Y(I) - Y1) * 150 / YD)
5144  IF X > 279 OR X < 13 OR Y < 0 OR Y > 154 THEN 5149
5145  IF I = 1 THEN 5148
5146  IF FILL$ <  > "Y" THEN 5148
5147  HPLOT  TO X,Y
5148  HPLOT X,Y
5149  NEXT I
5150  INVERSE
5151  PRINT "-";T$;"-";: NORMAL : PRINT  SPC( 28 -  LEN (T$))"CONTINUE?";
      : CALL  - 756
5152  PRINT : PRINT : PRINT "ANY CHANGES IN SCALE FACTORS (Y/N) ?";: INPUT
      "";A$
5153  IF A$ <  > "Y" THEN  RETURN
5154  GOTO 5120
```

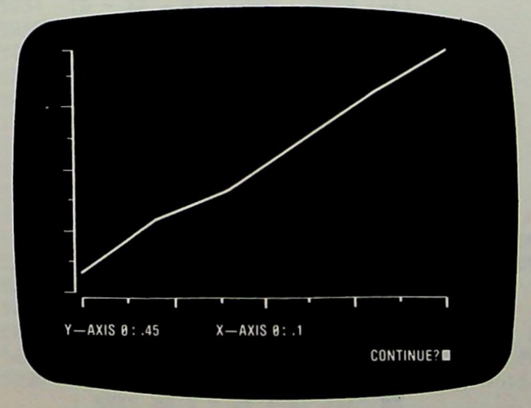
Fig. 2-5. Listing for PCARTXY subroutine.


```
NUMBER OF DATA POINTS ?  6

POINT 1    X=0          Y=.032
POINT 2    X=.02        Y=.135
POINT 3    X=.04        Y=.187
POINT 4    X=.06        Y=.268
POINT 5    X=.08        Y=.359
POINT 6    X=.1         Y=.435

TITLE ?  EXAMPLE DATA

MIN,MAX X VALUES ARE:  0-->.1
MIN,MAX Y VALUES ARE:  .032-->.435

MIN,MAX X-AXIS SCALE:  ?  0,.1
MIN,MAX Y-AXIS SCALE:  ?  0,.45

FILL BETWEEN POINTS ?  N
```

(A) Output results.



(B) Video plot—six discrete points.



(C) Video plot—points connected.

Fig. 2-6. Plotted data points.

driver program of Fig. 2-2, except that the GOSUB 5000 statement in line 155 is now changed to GOSUB 5100. When the PCARTXY subroutine is executed, you can choose to plot either the individual data points, or have the points connected by straight line segments when asked to respond to the question, FILL BETWEEN POINTS.

Example 2-2

Graph the following 6 discrete (X,Y) points: (0, 0.032), (0.02, 0.135), (0.04, 0.187), (0.06, 0.268), (0.08, 0.359), and (0.1, 0.435). Fig. 2-6A shows the results after this data is entered. The X-axis varies from 0 to 0.1 as Y ranges from 0.032 to 0.435. Fig. 2-6B shows the plot of the 6 discrete points, while Fig. 2-6C shows the same points connected together by straight line segments.

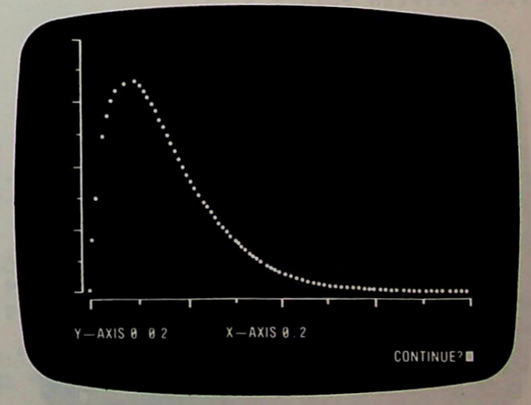
On the other hand, suppose we are interested in graphing continuous mathematical functions instead of discrete data points. The following example illustrates how this is accomplished.

Example 2-3

Suppose the transient response of a given network has the known time varying solution:

```
MIN,MAX VALUES OF T ?  0,2

TITLE ? Y=2.3*T*EXP(-5*T)

MIN,MAX X VALUES ARE:  0-->1.99245285
MIN,MAX Y VALUES ARE:  0-->.169194797

MIN,MAX X-AXIS SCALE:  ?  0,2
MIN,MAX Y-AXIS SCALE:  ?  0,.2

FILL BETWEEN POINTS ?  N
```

(A) Output results.



(B) Video plot.

Fig. 2-8. T plotted—minimum and maximum values.

```
100  REM     PCARTXY FUNCTION PLOT
101  HOME : TEST : VTAB 10: DIM X(265),Y(265)
102  N = 265 : I = 1
103  INPUT "MIN,MAX VALUES OF T ? ";A,B
104  C = (B - A) / 265
105  FOR T = A TO B STEP C
106  Y = 2.3 * T * EXP ( - 5 * T ) : REM  FUNCTION
107  X(I) = T : Y(I) = Y : I = I + 1
108  NEXT T : PRINT
109  INPUT "TITLE ? ";T$
110  GOSUB 5100 : TEXT : HOME : END
```

Fig. 2-7. Main program for plotting a function of time.

$$y(t) = 2.3te^{-5t}$$

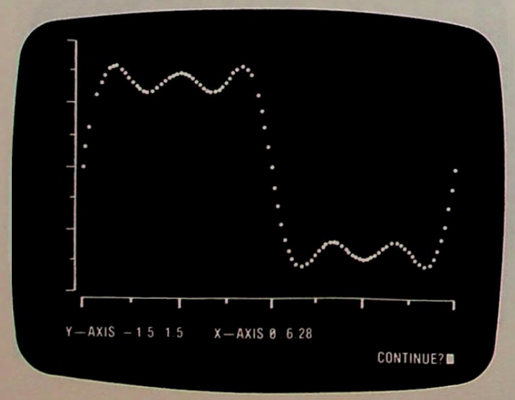
Fig. 2-7 lists the main program used to plot this function, which is defined in line 106 in terms of the variables, y, and time, t (the independent variable). Line 107 transforms these two variables to the array variables Y(I) and X(I) respectively, which are required by the PCARTXY subroutine.

```
100  REM     PCARTXY FUNCTION PLOT
101  HOME : TEST : VTAB 10: DIM X(265),Y(265)
102  N = 265 : I = 1
103  INPUT "MIN,MAX VALUES OF T ? ";A,B
104  C = (B - A) / 265
105  FOR T = A TO B STEP C
106  Y = 1.273 * SIN (T) + .424 + SIN (3 * T)
     + .255 * SIN (5 * T)
107  X(I) = T : Y(I) = Y : I = I + 1
108  NEXT T : PRINT
109  INPUT "TITLE ? ";T$
110  GOSUB 5100 : TEXT : HOME : END
```

(A) Main program.

```
MIN,MAX VALUES OF T ?  0, 6.28

TITLE ? FOURIER SQUARE WAVE

MIN,MAX X VALUES ARE:  0-->6.25630193
MIN,MAX Y VALUES ARE:  -1.18799496-->1.18798466

MIN,MAX X-AXIS SCALE:  ?  0, 6.28
MIN,MAX Y-AXIS SCALE:  ?  -1.5, 1.5

FILL BETWEEN POINTS ?  N
```

(B) Output results.



(C) Video plot.

Fig. 2-9. Plotting a function using PCARTXY.


```
5200  REM  SEMILOGARITHMIC PLOT SUBROUTINE FOR VIDEO DISPLAY (PSEMILOG)
5201  REM  X(I)=HORIZ.AXIS
5202  REM  Y(I)=VERTICAL AXIS
5203  REM  N=NO.OF DATA PTS.
5204  REM  T$=GRAPH TITLE
5205  REM  DETERMINE MINIMUM AND
5206  REM  MAXIMUM VALUES OF X & Y
5207  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1):Z9 = 1
5208  FOR I = 2 TO N
5209  IF (Y1 - Y(I)) <  = 0 THEN 5212
5210  Y1 = Y(I)
5211  GOTO 5214
5212  IF (Y2 - Y(I)) >  = 0 THEN 5214
5213  Y2 = Y(I)
5214  IF (X1 - X(I)) <  = 0 THEN 5217
5215  X1 = X(I)
5216  GOTO 5219
5217  IF (X2 - X(I)) >  = 0 THEN 5219
5218  X2 = X(I)
5219  NEXT I
5220  Y3 = Y1:Y4 = Y2: PRINT
5221  PRINT : PRINT "MIN,MAX X VALUES ARE:  ";X1;"-->";X2
5222  PRINT "MIN,MAX Y VALUES ARE:  ";Y3;"-->";Y4
5223  PRINT "MIN,MAX Y-AXIS SCALE: ? ";
5224  XY =  LOG (X1) /  LOG (10):XU =  LOG (X2) /  LOG (10):XS = X2 / X1:A
      A =  LOG (XS) /  LOG (10):AB =  INT (AA)
5225  CX =  INT ( ABS (XY - XU)) + 1
5226  IF  ABS (AE - XU) -  INT ( ABS (AE - XU)) < 1E - 4 THEN CX = CX - 1
5227  AE =  LOG (X1) /  LOG (10):AF =  INT (AE): IF (AE - AF) <  = .95 THEN
      5229
5228  XL = 10 ^ AE: GOTO 5231
5229  XL = 10 ^ AF
5230  AC =  INT (AE):XL = 10 ^ AC
5231  IF XL * (10 ^ CX) < X2 THEN CX = CX + 1
5232  IF CX > 4.9 THEN  HOME : VTAB 12: INVERSE : PRINT "DATA EXCEEDS 4 C
      YCLES CANNOT PLOT GRAPH": NORMAL : END
5233  INPUT "";Y1,Y2
5234  YD = Y2 - Y1: HOME : VTAB (24)
5235  INPUT "FILL BETWEEN POINTS? ";FILL$
5236  PRINT : PRINT
5237  REM  PLOT AXES WITH SCALES
5238  REM  ALL SAME Y OR X ?
5239  IF CX = 0 THEN CX = 1
5240  IF YD = 0 THEN YD = 1
5241  HCOLOR= 7: HGR
5242  REM  VERTICAL AXIS--------
5243  HPLOT 8,0 TO 8,150
5244  FOR Y = 150 TO 0 STEP  - (150 / 8): HPLOT 5,Y TO 7,Y: NEXT Y
5245  FOR Y = 150 TO 0 STEP  - (150 / 4): HPLOT 3,Y TO 5,Y: NEXT Y
5246  REM   HORIZ AXIS ----------
5247  HPLOT 13,154 TO 279,154
5248  U = 266 / CX: FOR X = 13 TO 279 STEP U: HPLOT X,155 TO X,160: NEXT X
5249  RT = 266 / CX:W = 0: FOR MM = 1 TO CX: FOR PZ = 2 TO 10 STEP 2
5250  LQ = (( LOG (PZ) /  LOG (10)) * RT) + 13 + W: HPLOT LQ,155 TO LQ,157
      : NEXT PZ:W = RT + W: NEXT MM
5251  VTAB 24: PRINT "MIN,MAX  Y:";Y3;",";Y4;"  X:";X1;",";X2
5252  PRINT "SCALE  Y=";Y1;":";Y2;"  X=";XL;":";XL * 10 ^ CX
```

Continued on next page.

Fig. 2-10. Listing for PSEMILOG subroutine.

```
5253  Z = 0
5254  FOR G = 0 TO CX7:H(G) = XL * 10 ^ G: NEXT G
5255  :
5256  FOR I = 1 TO N
5257  X = 13 + (( LOG (X(I) / XL)) * 266 / ( LOG (10) * CX))
5258  Y = 150 - ((Y(I) - Y1) * 150 / YD)
5259  IF X > 279 OR X < 13 OR Y < 0 OR Y > 154 THEN 5264
5260  IF I = 1 THEN 5263
5261  IF FILL$ < > "Y" THEN 5263
5262  HPLOT  TO X,Y
5263  HPLOT X,Y
5264  NEXT I
5265  INVERSE
5266  PRINT "-";T$;"-";: NORMAL : PRINT  SPC( 28 -  LEN (T$))"CONTINUE?";
      : CALL  - 756
5267  PRINT : PRINT : PRINT "ANY CHANGES IN SCALE FACTORS (Y/N) ?";: INPUT
      "";A$
5268  IF A$ < > "Y" THEN  RETURN
5269  GOTO 5221
```

Fig. 2-10 (cont). Listing for PSEMILOG subroutine.

Since most functions are single-valued, there will be 265 continuous points, which are recognized by the DIM statement in line 101. When this main program is run, we are first asked to select the minimum and maximum values of the independent variable (t) over which this function is to be plotted, after which we are asked for a suitable title for our graph.

For this example, we are interested in how this function appears as time varies from 0 to 2.0 seconds. When executed, we find that y ranges from 0 to 0.17 as t varies from 0 to 2.0, as shown in Fig. 2-8A, while the resulting plot is shown in Fig. 2-8B. The graph remains on the screen until the RETURN key is pressed, after which we are asked whether or not we wish to change the scale limits of either one or both axes.

Periodic, or circular functions, such as those which have sine or cosine terms are easily graphed on the display, as illustrated by the following example.

Example 2-4

Plot the first three terms for the Fourier series of a square wave, such that:

$$f(x) = 1.273\sin(x) + 0.424\sin(3x) + 0.255\sin(5x)$$

Fig. 2-9A lists the main program used, plotting the function in terms of the independent variable x, which is chosen to range from 0 to 6.28 radians. Fig. 2-9B shows the output results along with the plot of this function in Fig. 2-9C.

SEMILOGARITHMIC PLOTS

The PSEMILOG subroutine, listed in Fig. 2-10, permits the plotting of points on a semilogarithmic coordinate system, which is logarithmic in X and linear in Y. Depending on the range of values for the horizontal axis, PSEMILOG automatically scales the X-axis for values with a maximum of

```
100  REM  SAMPLE MAIN PROGRAM
105  DIM X(265),Y(265):SP$ = " "
110  TEXT : HOME : INPUT "NUMBER OF DATA POINTS? ";N
115  PRINT : PRINT : FOR I = 1 TO N
116  IF I > 9 THEN SP$ = ""
120  PRINT "POINT ";SP$;I;
125  INPUT "   X = ";X(I)
130  CALL  - 998
135  HTAB 26
140  INPUT "Y = ";Y(I)
145  NEXT I
150  PRINT : INPUT "TITLE? ";T$
155  PRINT : GOSUB 5200: TEXT : HOME
160  END : STOP : END : STOP
```

Fig. 2-11. Simple main program for PSEMILOG.

(A) Output results.

```
NUMBER OF DATA POINTS TO BE PLOTTED ?  15

POINT 1    X=.001      Y=1.49
POINT 2    X=.002      Y=1.48
POINT 3    X=.008      Y=1.43
POINT 4    X=.01       Y=1.40
POINT 5    X=.025      Y=1.29
POINT 6    X=.05       Y=1.15
POINT 7    X=.088      Y=0.95
POINT 8    X=.11       Y=.85
POINT 9    X=.3        Y=.33
POINT 10   X=.5        Y=.12
POINT 11   X=.94       Y=.013
POINT 12   X=1.23      Y=.01
POINT 13   X=4.3       Y=.099
POINT 14   X=7.1       Y=.098
POINT 15   X=9.2       Y=.098

MIN,MAX X VALUES ARE:  1E-3-->9.2
MIN,MAX Y VALUES ARE:  .01-->1.49

MIN,MAX Y-AXIS SCALE:  ?  0,1.5
```

(B) Video plot.

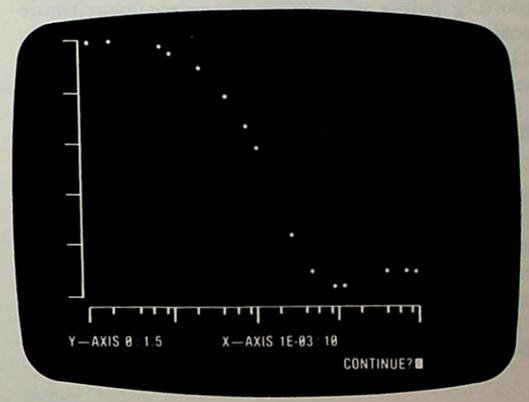


Fig. 2-12. Number of data points plotted.

4 cycles. However, the user is still able to set the range for the Y-axis scale.

Example 2-5

Plot the following 15 X-Y data points: (0.001, 1.49), (0.002, 1.48), (0.008, 1.43), (0.01, 1.40), (0.025, 1.29), (0.05, 1.15), (0.088, 0.95), (0.11, 0.85), (0.3, 0.33), (0.5, 0.12), (0.94, 0.013), (1.23, 0.01), (4.3, 0.099), (7.1, 0.098), and (9.2, 0.098). Using the driver program of Fig. 2-11, the resulting scale limits and graph for these 15 points are shown in Fig. 2-12. The graph, which shows 4 cycles, remains on the screen until the RETURN key is pressed. Like the PCARTXY subroutine, PSEMILOG allows you to connect straight line segments between consecutive points.

Continuous functions that require a semilogarithmic coordinate system (such as magnitude and phase plots of transfer functions, or Bode plots) are also easily graphed using PSEMILOG.

Fig. 2-13 lists a main program that can be used with PSEMILOG to plot, for example, the magnitude response:

$$dB = -10\ \log(1 + \omega^4)$$

The magnitude function is defined in line 108. Line 109 transforms the variables T (i.e., ω) and Y to the arrays X(I) and Y(I) respectively which are required by the PSEMILOG subroutine.

Fig. 2-14 shows the magnitude plot for the transfer function as frequency is chosen to vary

```
100  REM    SAMPLE MAIN PROGRAM
101  HOME : TEST : VTAB 10: DIM X(265),Y(265)
102 N = 265 : I = 1
103  INPUT "MIN,MAX VALUES OF T ? ";A,B
104 D = LOG (B / A) / LOG (10)
105 D = INT (D + .5)
106  FOR F = 0 TO D - 1
107  FOR T = A * 10 ^ F TO (B / 10 ^  (D - F - 1)) +
     A / 10 ^ F STEP (A * 10 ^ F ) * (D) / 10
108 Y = - 10 * LOG (1 + (T ^ 4)) / LOG (10)
109 X(I) = T : Y(I) = Y : I = I + 1
110  NEXT T
111  NEXT F
112  N = N - 1
113  GOSUB 5200 : TEXT : HOME : END
```

Fig. 2-13. Simple main program for plotting magnitude using PSEMILOG.

from 0.1 to 10 Hz, or a range of two decades (cycles).

LOG-LOG PLOTS

Fig. 2-15 lists the LOGLOG subroutine used for plotting points whose range requires logarithmic scales for both axes. The main program of Fig. 2-11 can be used except that the GOSUB 5200 statement is changed to GOSUB 5300 in line 155.

Example 2-6

Graph the following 5 discrete (X,Y) points: (0.00075, 0.457), (0.003, 0.349), (0.008, 0.191), (0.015, 0.083), and (0.052, 0.01). Using the driver program of Fig. 2-11 and the GOSUB 5300 statement in line 155, the resulting scale limits and graph for these 5 points are shown in Fig. 2-16 which shows that the X-axis has 3 cycles while the Y-axis has 2 cycles.

POLAR PLOTS

Polar plots are frequently used in field theory and control systems, where the plot describes both a radial distance as well as the angle (as measured counterclockwise from the horizontal) associated with it. Examples are the electric field distribution of an antenna, or Nyquist plots of the stability of feedback systems.

The VPOLAR program shown in Fig. 2-17 permits the plotting of either discrete data points or functions expressed in polar coordinates. The polar function must be defined in line 103. The variable I is in degrees and is converted to radians by dividing it by 57.29577. As examples, Table 2-2 lists the polar forms of a variety of frequently used functions, where the angle is expressed in radians.

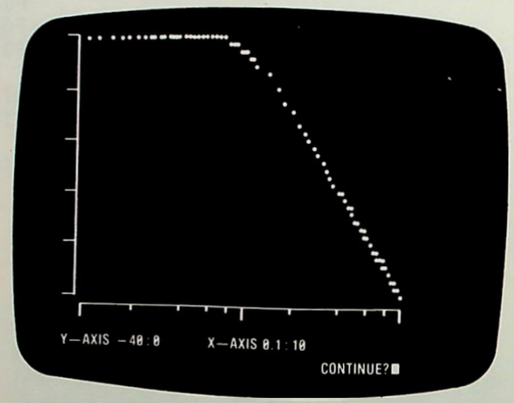


Fig. 2-14. Magnitude plot for the transfer function.

After the data points are entered or the polar function is determined, the program asks for a graph title, and then determines the maximum radius that exists for the discrete data points entered or the function that was defined. We are then asked to enter a value for the radius which is the maximum scales for both axes. The plot is then shown on the video display. In addition, the title, maximum radius, X-axis and Y-axis increments are displayed. In line 105 the variable RAD is defined which is used to represent the angle in terms of radians.

Table 2-2. Common Polar Functions

Name	Equation
Circle:	$r = A\cos\theta$ or $r = \lvert A\sin\theta\rvert$
Cardioid:	$r = A(1 + \cos\theta)$ or $r = \lvert A(1 + \sin\theta\rvert$
Lemniscate:	$r = \lvert A\cos(n\theta)\rvert$ or $r = \lvert A\sin(n\theta)\rvert$
Limacon:	$r = \lvert A + B\cos\theta\rvert$ or $r = \lvert A + B\sin\theta\rvert$
N-Leaved Rose:	$r = \lvert A\cos(n\theta)\rvert$ or $r = \lvert A\sin(n\theta)\rvert$
Archimedian Spiral:	$r = A\theta$
Hyperbolic Spiral:	$r = A/\theta$
Logarithmic Spiral:	$r = \exp(b\theta)$


```
5300  REM  LOG-LOG PLOT SUBROUTINE FOR VIDEO DISPLAY (LOGLOG)
5301  REM  X(I)=HORIZ.AXIS
5302  REM  Y(I)=VERTICAL AXIS
5303  REM  N=NO.OF DATA PTS.
5304  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
5305  FOR I = 2 TO N
5306  IF (Y1 - Y(I)) <  = 0 THEN 5309
5307  Y1 = Y(I)
5308  GOTO 5311
5309  IF (Y2 - Y(I)) >  = 0 THEN 5311
5310  Y2 = Y(I)
5311  IF (X1 - X(I)) <  = 0 THEN 5314
5312  X1 = X(I)
5313  GOTO 5316
5314  IF (X2 - X(I)) >  = 0 THEN 5316
5315  X2 = X(I)
5316  NEXT I
5317  XS = X2 / X1:AA =  LOG (XS) /  LOG (10):AB =  INT (AA):XY =  LOG (X1
      ) /  LOG (10):XU =  LOG (X2) /  LOG (10):YY =  LOG (Y1) /  LOG (10):
      YU =  LOG (Y2) /  LOG (10)
5318  CX =  INT ( ABS (XY - XU)) + 1
5319  IF  ABS (AE - XU) -  INT ( ABS (AE - XU)) < 1E - 4 THEN CX = CX - 1
5320  AE =  LOG (X1) /  LOG (10):AF =  INT (AE): IF (AE - AF) <  = .95 THEN
      5322
5321  XL = 10 ^ AE: GOTO 5324
5322  XL = 10 ^ AF
5323  AC =  INT ( LOG (X1) /  LOG (10)):XL = 10 ^ AC
5324  IF XL * (10 ^ CX) < X2 THEN CX = CX + 1
5325  IF CX > 4.9 THEN  HOME : VTAB 12: INVERSE : PRINT "DATA EXCEEDS 4 C
      YCLES. CANNOT PLOT GRAPH": NORMAL : END
5326  YS = Y2 / Y1:BA =  LOG (YS) /  LOG (10):BB =  INT (BA)
5327  CY =  INT ( ABS (YY - YU)) + 1
5328  IF  ABS (BE - YU) -  INT ( ABS (BE - YU)) < 1E - 4 THEN CY = CY - 1
5329  BE =  LOG (Y1) /  LOG (10):BF =  INT (BE): IF (BE - BF) <  = .95 THEN
      5331
5330  YL = 10 ^ BE: GOTO 5333
5331  YL = 10 ^ BF
5332  BC =  INT (BE):YL = 10 ^ BC
5333  IF YL * (10 ^ CY) < Y2 THEN CY = CY + 1
5334  IF CY > 4.9 THEN  HOME : VTAB 12: INVERSE : PRINT "DATA EXCEEDS 4 C
      YCLES. CANNOT PLOT GRAPH": NORMAL : END
5335  HOME : VTAB 24: INPUT "FILL BETWEEN POINTS ? ";FILL$
5336  PRINT : PRINT
5337  REM  ALL SAME Y,X ?
5338  IF CX = 0 THEN CX = 1
5339  IF CY = 0 THEN CY = 1
5340  REM  PLOT AXES WITH SCALES
5341  HCOLOR= 7: HGR
5342  REM  VERTICAL AXIS--------
5343  HPLOT 8,0 TO 8,150
5344  FOR Y = 150 TO 0 STEP  - (150 / CY): HPLOT 3,Y TO 7,Y: NEXT Y
5345  RT = 150 / CY:W = 0
5346  FOR MM = 1 TO CY: FOR PZ = 2 TO 10 STEP 2:LQ = 150 - (( LOG (PZ) /
      LOG (10)) * RT) + W: HPLOT 5,LQ TO 6,LQ: NEXT PZ:W = W - RT: NEXT M
      M
5347  REM   HORIZ AXIS ----------
5348  HPLOT 13,154 TO 279,154
5349  U = 266 / CX: FOR X = 13 TO 279 STEP U: HPLOT X,155 TO X,160: NEXT X
```

Continued on next page.

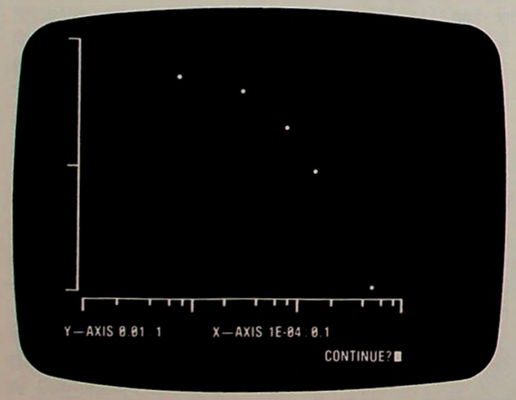
Fig. 2-15. Listing for LOGLOG subroutine.

```
5350  RT = 266 / CX:W = 0
5351  FOR MM = 1 TO CX: FOR PZ = 2 TO 10 STEP 2:LQ = (( LOG (PZ) /  LOG (
      10)) * RT) + 13 + W: HPLOT LQ,155 TO LQ,157: NEXT PZ:W = RT + W: NEXT
      MM
5352  :
5353  VTAB 24: PRINT "MIN,MAX  Y:";Y1;",";Y2;"  X:";X1;",";X2
5354  PRINT "SCALE  Y=";YL;":";YL * 10 ^ CY;"  X=";XL;":";XL * 10 ^ CX
5355  FOR I = 1 TO N
5356  X = 13 + (( LOG (X(I) / XL)) * 266 / ( LOG (10) * CX))
5357  Y = 150 - (( LOG (Y(I) / YL)) * 150 / (CY *  LOG (10)))
5358  IF X > 279 OR X < 13 OR Y < 0 OR Y > 154 THEN 5363
5359  IF I = 1 THEN 5362
5360  IF FILL$ <  > "Y" THEN 5362
5361  HPLOT  TO X,Y
5362  HPLOT X,Y
5363  NEXT I
5364  INVERSE
5365  PRINT "-";T$;"-";: NORMAL : PRINT  SPC( 28 -  LEN (T$))"CONTINUE?";
      : CALL  - 756
5366  PRINT : RETURN
```

Fig. 2-15 (cont). Listing for LOGLOG subroutine.

```
NUMBER OF DATA POINTS TO BE PLOTTED ?  5

POINT 1    X=.00075    Y=.457
POINT 2    X=.003      Y=.349
POINT 3    X=.008      Y=.191
POINT 4    X=.015      Y=.083
POINT 5    X=.052      Y=.01

MIN,MAX X VALUES ARE:  7.5E-04-->.052
MIN,MAX Y VALUES ARE:  .01-->.457
```

(A) Output results.



(B) Video plot.

Fig. 2-16. Five data points plotted.

Example 2-7

Plot the cardioid

$$r = 2(1 + \cos(\theta))$$

so that there will be a point plotted every 4 degrees. Line 103 should then read:

```
103 R(K) = 2 * (1 + COS (I / RAD))
```

When run, the maximum radius (or the maximum value of the function) was found to be 4.0, giving the results shown in Fig. 2-18. Other polar func-


```
100  REM  POLAR PLOT FOR VIDEO DISPLAY (VPOLAR)
101  LOMEM: 16384: GOTO 105
102  REM  -- ACTUAL FUNCTION --
103  R(K) = 2 * (1 +  COS (I / RAD))
104  RETURN
105  RAD = 57.29577: REM   RADIANS
106  TEXT
107  HOME : VTAB (6): PRINT  TAB( 4): INVERSE : PRINT "POLAR PLOTTING FOR
     VIDEO DISPLAY": NORMAL
108  VTAB 20:Z9 = 1
109  PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
110  VTAB (13): PRINT  SPC( 11)"APPLESOFT VERSION"
111  FOR II = 1 TO 2500: NEXT
112  DIM X(361),Y(361),R(361)
113  HOME : PRINT : PRINT "POLAR FUNCTION IS DEFINED IN LINE 103"
114  PRINT : PRINT "FOR EXAMPLE:"
115  PRINT "103 R(K) = 2 * (1 + COS (I/RAD))"
116  PRINT : PRINT "(EXAMPLE DESCRIBES A CARDIOID)       "
117  PRINT "--> YOU MAY USE 'RAD' FOR 57.29578  "
118  VTAB (12): PRINT "GRAPH IS FOR:": PRINT
119  PRINT  SPC( 12)"1)  DISCRETE DATA POINTS": PRINT : PRINT  SPC( 12)"2
     )  A POLAR FUNCTION"
120  VTAB (23): INPUT "YOUR CHOICE? ";II
121  IF II < 1 OR AA > 2 THEN 120
122  HOME : VTAB (12): PRINT "ENTER TITLE: ";: INPUT "";T$
123  IF  LEN (T$) > 20 THEN 122
124  IF II = 2 THEN 168
125  VTAB (16): INPUT "NUMBER OF DATA POINTS ? ";N
126  PRINT : FOR I = 1 TO N: PRINT "POINT #";I;": ";: INPUT "MAGNITUDE, A
     NGLE? ";R(I),Z
127  A = Z / RAD:X(I) = R(I) *  COS (A):Y(I) = R(I) *  SIN (A): NEXT
128  Z9 = 2: GOTO 177
129  PRINT : PRINT : PRINT "DO YOU WANT ME TO FILL IN THE SPACES": PRINT
     : INPUT "BETWEEN THE POINTS (Y/N) ";FILL$
130  PRINT : PRINT : PRINT : HGR : HCOLOR= 7
131  S = 4:T = 50
132  HPLOT 140,4 TO 140,155: HPLOT 50,80 TO 229,80
133  FOR II = 1 TO 11
134  HPLOT 138,S TO 139,S:S = S + 152 / 10
135  HPLOT T,78 TO T,79:T = T + 180 / 10
136  NEXT II
137  VTAB (24): INVERSE
138  PRINT "-";T$;"-";: NORMAL : PRINT " RADIUS = ";RM
139  PRINT
140  PRINT "SCALE = "; INT ((RM / 5) * 100 + .5) / 100;" PER DIVISION ";
141  FOR K = 0 TO N
142  X = 140 + (X(K) * 89 / RM):Y = 80 - (Y(K) * 76 / RM)
143  IF X > 278 OR Y > 158 THEN 149
144  IF X < 0 OR Y < 0 THEN 149
145  IF Z9 = 2 AND K = 1 AND FILL$ = "Y" THEN 148
146  IF K = 0 THEN 148
147  IF FILL$ = "Y" THEN  HPLOT  TO X,Y: GOTO 149
148  HPLOT X,Y
149  NEXT K
150  PRINT  TAB( 30);"CONTINUE?";: CALL  - 756
151  TEXT : HOME : VTAB (8)
152  PRINT  SPC( 5)"1)  CHANGE AXIS SCALES"
153  PRINT
```

Continued on next page.

Fig. 2-17. Listing for VPOLAR program.

```
154  IF Z9 <  > 2 THEN 158
155  PRINT  SPC( 5)"2)  QUIT": PRINT : PRINT : INPUT "YOUR CHOICE? ";Z8
156  ON Z8 GOTO 161,166
157  GOTO 151
158  PRINT  SPC( 5)"2)  PLOT INCREMENT": PRINT : PRINT  SPC( 5)"3)  BOTH"
     : PRINT : PRINT  SPC( 5)"4)  QUIT": PRINT : PRINT
159  INPUT "WHICH ? ";Z7
160  PRINT : ON Z7 GOTO 161,164,165,166
161  INPUT "MAX RADIUS ";RM
162  IF RM >  = R2 THEN 129
163  INVERSE : PRINT : PRINT "PLOT RADIUS MUST BE AT LEAST ";R2: NORMAL :
     GOTO 161
164  Z9 = 0: GOTO 168
165  Z9 = 1: GOTO 168
166  PRINT
167  PRINT : END
168  K = 0
169  PRINT : INPUT "ENTER THE PLOT INCREMENT (1 TO 360) ?";G
170  HOME : VTAB (12): PRINT  TAB( 12)"COMPUTING DEGREES  ":
171  FOR I = 0 TO 361 STEP G: VTAB (14): PRINT  TAB( 20)I
172  GOSUB 103
173  IF K > 360 / G THEN 176
174  A = I / RAD:X(K) = R(K) *  COS (A):Y(K) = R(K) *  SIN (A)
175  K = K + 1: NEXT I
176  N = K - 1: REM  MIN/MAX
177  R1 = R(1):R2 = R(1)
178  FOR I = 2 TO N
179  IF (R1 - R(I)) <  = 0 THEN 181
180  R1 = R(I): GOTO 183
181  IF (R2 - R(I)) >  = 0 THEN 183
182  R2 = R(I): IF R2 < 1 THEN R2 = 1
183  NEXT I
184  IF Z9 = 0 THEN 129
185  HOME
186  PRINT "THE DATA HAS A MAXIMUM RADIUS OF ";R2
187  PRINT
188  INPUT "DESIRED MAX RADIUS FOR THE AXES ? ";RM
189  IF RM >  = R2 THEN 191
190  PRINT : INVERSE : PRINT "PLOT RADIUS MUST BE AT LEAST ";R2: NORMAL :
     PRINT : GOTO 188
191  GOTO 129
```

Fig. 2-17 (cont). Listing for VPOLAR program.

tions plotted using the VPOLAR routine are shown in Fig. 2-19.

PLOTTING WITH A LINE PRINTER

For those of you who have a line printer, you would probably like to make a hard-copy record of a particular graph either for inclusion in a report, or for later comparison with other results. The following programs allow continuous functions to be graphed on virtually all serial and parallel line printers used with the Apple II. For all these programs, the printer interface card is set at SLOT = 1. If a parallel printer is used, 80 columns maximum will be printed.

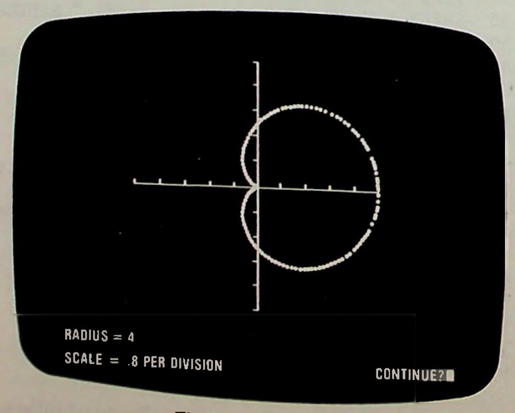
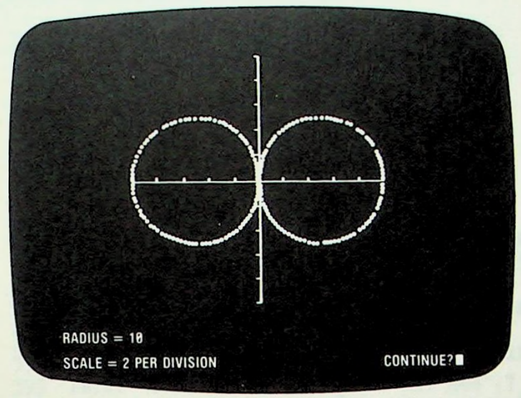
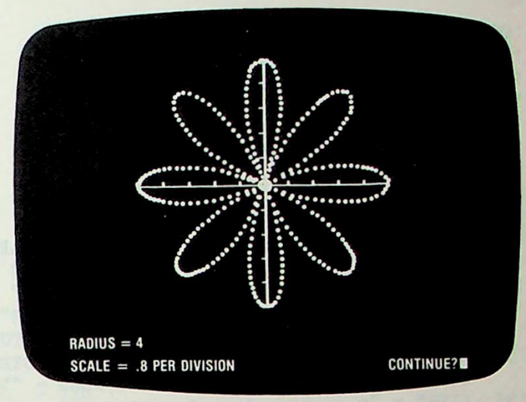


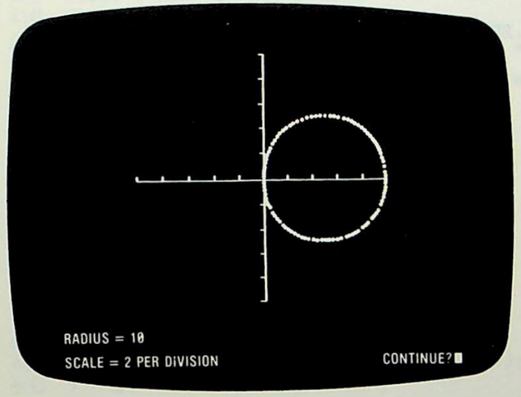
Fig. 2-18. Cardioid plot.



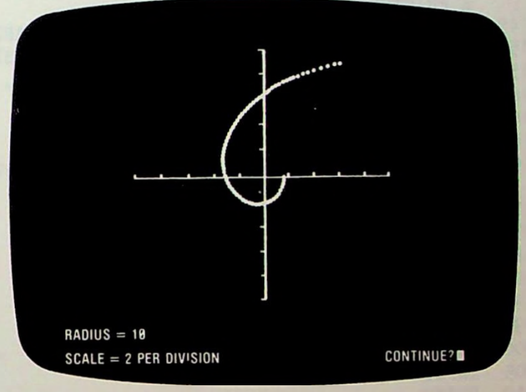
(A) Lemniscate.



(B) 6-leaved rose.



(C) Circle.



(D) Hyperbolic spiral.

(E) Archimedean spiral.

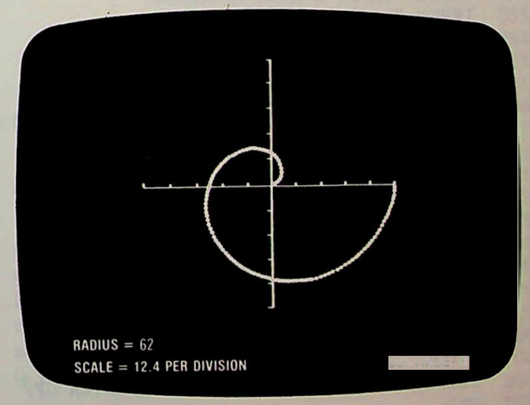


Fig. 2-19. Video plot of various polar functions.

```
100  REM ***** LINEPLOT *****
105  REM * X-Y PLOT ROUTINE *
106  REM * FOR LINE PRINTER *
107  REM ********************
108  :
109  SLOT = 1: REM   PRINTER SLOT
110  :
111  GOTO 135
112  PRINT  CHR$ (4);"PR#";SL
113  PRINT  CHR$ (9);"80N": REM   PARALLEL CARD SET TO 80 COL.
114  RETURN : REM DEL LINE 113       IF USING SERIAL CARD
115  IF XX < 0 THEN XX =  ABS (XX):NEG$ = "-"
120  IF XX > 999999.99 THEN  PRINT "TOO LARGE": END
125  IF XX < .005 THEN Z$ = ".00":LG = 3: RETURN
130  XX =  INT (XX * 100 + .5) / 100:Z$ =  STR$ (XX + .005):Z$ =  LEFT$ (Z
     $, LEN (Z$) - 1):Z$ = NEG$ + Z$:NEG$ = "":LG =  LEN (Z$): RETURN
135  LINE$ = "+----+----+----+----+----+----+----+----+----+----+"
140  HOME : VTAB (4): HTAB (4): INVERSE : PRINT "X-Y PLOT PROGRAM FOR LIN
     E PRINTER": NORMAL
145  VTAB (7): PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
150  VTAB (10): PRINT  SPC( 11)"APPLESOFT VERSION"
155  FOR II = 1 TO 2500: NEXT II
160  DIM X(100),Y(100)
161  HOME
162  INVERSE
165  PRINT "FUNCTION MUST BE DEFINED IN LINE 275  ": PRINT "EXAMPLE>  275
     Y=EXP(-.2 * X) * SIN(X)": PRINT
170  NORMAL
175  PRINT "INPUT THE MINIMUM 'X' VALUE OF"
180  PRINT "THE FUNCTION TO BE PLOTTED ";
185  INPUT "       ";X1
190  PRINT
195  PRINT "INPUT THE MAXIMUM 'X' VALUE OF"
200  PRINT "THE FUNCTION TO BE PLOTTED ";
205  INPUT "       ";X2
210  PRINT
215  PRINT "INPUT X AXIS INCREMENT ";
220  INPUT "           ";R
225  PRINT : PRINT
230  INPUT "X AXIS TITLE? ";X$
240  PRINT
245  INPUT "Y AXIS TITLE? ";Y$
250  K =  LEN (X$)
255  I = 1:XD = X2 - X1:S = XD / R
265  FOR X = X1 TO X2 STEP R
270  REM  INSERT FUNCTION HERE
275  Y =  EXP ( - .2 * X) *  SIN (X)
280  REM  INSERT FUNCTION ABOVE
285  X(I) = X:Y(I) = Y:I = I + 1:N = I - 1
290  NEXT
295  GOSUB 310
300  GOSUB 385
301  PRINT  CHR$ (12): REM  FORM FEED
302  PRINT  CHR$ (4);"PR#0": REM  PRINTER OFF
303  END
310  REM  FIND MIN,MAX VALUE OF Y
315  Y1 = Y(1):Y2 = Y(1)
320  FOR I = 1 TO N
```

Fig. 2-20. Listing for LINEPLOT program.

Continued on next page.


```
325  IF Y1 - Y(I) <  = 0 THEN 340
330  Y1 = Y(I)
335  GOTO 350
340  IF Y2 - Y(I) >  = 0 THEN 350
345  Y2 = Y(I)
350  NEXT I
351  PRINT
355  PRINT "MINIMUM Y-AXIS VALUE = ";Y1
360  PRINT "MAXIMUM Y-AXIS VALUE = ";Y2
365  PRINT : INPUT "ENTER MINIMUM Y-AXIS VALUE ";Y1
370  INPUT "ENTER MAXIMUM Y-AXIS VALUE ";Y2
375  YD = Y2 - Y1: RETURN
380  REM  -----------------
385  REM  ### PRINTIT ###
390  HOME : VTAB (12)
395  PRINT "TURN ON PRINTER & HIT ANY KEY";: CALL  - 756
400  L = YD / 50:C1 = Y1 - L / 2:D1 = Y2 + L / 2:W = 1: HOME
405  PRINT : REM  DO NOT DELETE
410  GOSUB 112: REM START PRINTER
420  PRINT  TAB( 32)Y$: PRINT  CHR$ (10)
425  PRINT  SPC( 8)" ";
430  FOR U = 0 TO 5: PRINT  SPC( CC);U * 10 * L + Y1;
435  CC = 8
440  NEXT U: PRINT " "
445  PRINT  TAB( 11)LINE$
450  FOR I = 1 TO S + 1
455  IF Y(I) < C1 THEN 515
460  IF Y(I) >  = D1 THEN 515
465  XX = X(I): GOSUB 115
470  VLTR$ =  MID$ (X$,W,1)
475  IF VLTR$ = "" THEN VLTR$ = " "
480  PRINT VLTR$;" "; TAB( 7 - LG)Z$;
485  VV = (Y(I) - Y1) / L + 1.5
490  PRINT " I"; TAB( VV);"*"; TAB( 53 - VV);"I"
500  W = W + 1: NEXT I
505  PRINT  TAB( 11)LINE$
510  RETURN
515  XX = X(I): GOSUB 115
520  PRINT  MID$ (X$,W,1);"  ";Z$
525  PRINT  TAB( 9);"I"; TAB( 52)"I": GOTO 500
```

Fig. 2-20 (cont). Listing for LINEPLOT program.

Cartesian Plots

Fig. 2-20 shows the LINEPLOT program which permits the plotting of a function, which must be defined in line 275, on a Cartesian coordinate system. If a serial printer is used, delete line 113. The program asks for the minimum and maximum values of X for which the function is to be plotted, as well as the X-axis increment scale. In addition, the program will label both axes.

Example 2-8

Plot the function $y(t) = e^{-.2x}\sin(x)$ as x varies from 0 to 15 radians. Fig. 2-21 shows the resultant graph, for which the scale of the Y-axis was set to vary from −0.5 to 1.0.

Semilogarithmic Plots

The XLOGPLOT program shown in Fig. 2-22 permits plotting of a function having a logarithmic behavior, which must be defined in line 275. Such would be the case of the frequency response of a filter which might cover several decades. Line 435 must be deleted if a serial printer is used.

Example 2-9

Plot the frequency response of a second-order low-pass filter, having the form:

$$dB = -10\log(1 + f^2)$$

```
MINIMUM Y-AXIS VALUE = -.397434
MAXIMUM Y-AXIS VALUE =  .738963

ENTER MINIMUM Y-AXIS VALUE ? -.5
ENTER MAXIMUM Y-AXIS VALUE ? 1.0
```

(A) Output results.

```
                          AMPLITUDE

          -.5       -.2        .1        .4        .7        1
           +---------+---------+---------+---------+---------+
R    0.00 I                    *                             I
A    0.50 I                              *                   I
D    1.00 I                                        *         I
I    1.50 I                                         *        I
A    2.00 I                                    *             I
N    2.50 I                               *                  I
S    3.00 I                    *                             I
     3.50 I          *                                       I
     4.00 I     *                                            I
     4.50 I   *                                              I
     5.00 I     *                                            I
     5.50 I          *                                       I
     6.00 I               *                                  I
     6.50 I                    *                             I
     7.00 I                       *                          I
     7.50 I                        *                         I
     8.00 I                       *                          I
     8.50 I                      *                           I
     9.00 I                    *                             I
     9.50 I                  *                               I
    10.00 I                *                                 I
    10.50 I               *                                  I
    11.00 I               *                                  I
    11.50 I                *                                 I
    12.00 I                 *                                I
    12.50 I                 *                                I
    13.00 I                    *                             I
    13.50 I                    *                             I
    14.00 I                     *                            I
    14.50 I                    *                             I
    15.00 I                    *                             I
          +---------+---------+---------+---------+---------+
```

(B) Printer plot.

Fig. 2-21. Function $y(t) = e^{-.2x} \sin(x)$ plotted as x varies from 0 to 15 radians.

Fig. 2-23 shows the resultant plot from 0.1 to 10 Hz.

Polar Plots

The POLARPLT program shown in Fig. 2-24 permits the plotting of polar functions (Table 2-2), such as those describing cardioids, spirals, etc., and must be defined in line 190. Line 470 must be deleted if a serial printer is used.

When run, the program requires only a graph title and the scale limit for the two axes.

Example 2-10

Plot the N-leaved rose

$$r = \sin(3\theta)$$

As shown in Fig. 2-25, the scale limits of the two axes were arbitrarily set at 1.2.


```
100  REM  ***** XLOGPLOT ****
105  REM  ** SEMI-LOG PLOT **
106  REM  **   SUBROUTINE  **
107  REM  * FOR LINEPRINTER *
108  REM  ********************
109  :
115  SLOT = 1: REM   PRINTER SLOT
116  :
117  REM DEL LINE 435 IF YOU ARE
118  REM   USING THE SERIAL CARD
120  GOTO 145
125  IF XX < 0 THEN XX =  ABS (XX):NEG$ = "-"
130  IF XX > 999999.99 THEN  PRINT "TOO LARGE": END
135  IF XX < .005 THEN Z$ = ".00":LG = 3: RETURN
140  XX =  INT (XX * 100 + .5) / 100:Z$ =  STR$ (XX + .005):Z$ =  LEFT$ (Z
     $, LEN (Z$) - 1):Z$ = NEG$ + Z$:NEG$ = "":LG =  LEN (Z$): RETURN
145  LINE$ = "+----+----+----+----+----+----+----+----+----+----+"
150  HOME : VTAB (4): HTAB (4): INVERSE : PRINT "X-Y PLOT PROGRAM FOR LIN
     E PRINTER": NORMAL
155  VTAB (7): PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
160  VTAB (10): PRINT  SPC( 11)"APPLESOFT VERSION"
165  FOR II = 1 TO 2500: NEXT II
170  DIM X(100),Y(100)
175  HOME
180  INVERSE
185  PRINT "FUNCTION MUST BE DEFINED IN LINE 275  ": PRINT "EXAMPLE>  275
     Y=EXP(-.2 * X) * SIN(X)": PRINT
190  NORMAL
195  PRINT "INPUT THE MINIMUM 'X' VALUE OF"
200  PRINT "THE FUNCTION TO BE PLOTTED ";
205  INPUT "       ";X1
210  PRINT
215  PRINT "INPUT THE MAXIMUM 'X' VALUE OF"
220  PRINT "THE FUNCTION TO BE PLOTTED ";
225  INPUT "       ";X2
230  X1 =  LOG (X1) /  LOG (10):X2 =  LOG (X2) /  LOG (10)
235  PRINT
240  PRINT : PRINT
245  INPUT "X AXIS TITLE? ";X$
250  PRINT
255  INPUT "Y AXIS TITLE? ";Y$
260  K =  LEN (X$)
265  I = 1:XD = X2 - X1:S = XD / .1
270  FOR X = X1 TO X2 + .1 STEP .1
275  REM  INSERT FUNCTION HERE
280  Y = 20 *  LOG (1 /  SQR (1 + ((10 ^ X) ^ 2))) /  LOG (10)
285  REM  INSERT FUNCTION ABOVE
290  X(I) = X:Y(I) = Y:I = I + 1:N = I - 1
295  NEXT
300  GOSUB 320
305  GOSUB 400
310  PRINT  CHR$ (4);"PR#0": REM  PRINTER OFF
315  END
320  REM  FIND MIN,MAX VALUE OF Y
325  Y1 = Y(1):Y2 = Y(1)
330  FOR I = 1 TO N
335  IF Y1 - Y(I) <  = 0 THEN 350
340  Y1 = Y(I)
```

Continued on next page.

Fig. 2-22. Listing for XLOGPLOT program.

```
345  GOTO 360
350  IF Y2 - Y(I) >  = 0 THEN 360
355  Y2 = Y(I)
360  NEXT I
365  PRINT
370  PRINT "MINIMUM Y-AXIS VALUE = ";Y1
375  PRINT "MAXIMUM Y-AXIS VALUE = ";Y2
380  PRINT : INPUT "ENTER MINIMUM Y-AXIS VALUE ";Y1
385  INPUT "ENTER MAXIMUM Y-AXIS VALUE ";Y2
390  YD = Y2 - Y1: RETURN
395  REM  ------------------
400  REM   ### PRINT IT ###
405  HOME : VTAB (12): INVERSE
410  PRINT "TURN ON PRINTER & HIT ANY KEY";: CALL  - 756
415  NORMAL
420  L = YD / 50:C1 = Y1 - L / 2:D1 = Y2 + L / 2:W = 1: HOME
425  PRINT : REM  DO NOT DELETE
430  PRINT  CHR$ (4);"PR#"SLOT: REM   PRINTER SLOT
435  PRINT  CHR$ (9);"80N": REM  LINE WIDTH 80
440  PRINT  TAB( 32)Y$: PRINT  CHR$ (10)
445  PRINT  SPC( 8)" ";
450  FOR U = 0 TO 5: PRINT  SPC( CC);U * 10 * L + Y1;
455  CC = 8
460  NEXT U: PRINT " "
465  PRINT  TAB( 11)LINE$
470  FOR I = 1 TO S + 2
475  IF Y(I) < C1 THEN 535
480  IF Y(I) >  = D1 THEN 535
485  XX = 10 ^ X(I): GOSUB 125
490  VLTR$ =  MID$ (X$,W,1)
495  IF VLTR$ = "" THEN VLTR$ = " "
500  PRINT VLTR$;" "; TAB( 7 - LG)Z$;
505  VV = (Y(I) - Y1) / L + 1.5
510  PRINT " I"; TAB( VV);"*";
515  PRINT  TAB( 53 - VV);"I"
520  W = W + 1: NEXT I
525  PRINT  TAB( 11)LINE$
530  RETURN
535  XX = 10 ^ X(I): GOSUB 125
540  PRINT  MID$ (X$,W,1);"  ";Z$
545  PRINT  TAB( 9);"I"; TAB( 52)"I": GOTO 520
```

Fig. 2-22 (cont). Listing for XLOGPLOT program.

(A) Output results.

```
MIN, MAX X? 0.1, 10.0
MIN, MAX Y? -20, 5
X-AXIS TITLE ? LOG FREQUENCY IN HZ
Y-AXIS TITLE ? DB RESPONSE
```

```
                          DB RESPONSE

                  -20       -15       -10        -5         0         5
                   +----+----+----+----+----+----+----+----+----+----+----+
L      0.10 I                                                          *   I
O      0.13 I                                                          *   I
G      0.16 I                                                          *   I
       0.20 I                                                          *   I
F      0.25 I                                                         *    I
R      0.32 I                                                         *    I
E      0.40 I                                                         *    I
Q      0.50 I                                                        *     I
U      0.63 I                                                      *       I
E      0.79 I                                                    *         I
N      1.00 I                                                  *           I
C      1.26 I                                                *             I
Y      1.58 I                                              *               I
       2.00 I                                            *                 I
I      2.51 I                                         *                    I
N      3.16 I                                      *                       I
       3.98 I                                   *                          I
H      5.01 I                                 *                            I
Z      6.31 I                              *                               I
       7.94 I                         *                                    I
      10.00 I*                                                             I
        +----+----+----+----+----+----+----+----+----+----+----+----+----+
```

(B) Printer plot.

Fig. 2-23. Plotted frequency response of low-pass filter.

```
100  REM  ***** POLARPLT *****
101  REM  ** POLAR PLOTTING **
102  REM  * FOR LINE PRINTER *
103  REM  ********************
104  :
105  SLOT = 1: REM  PRINTER SLOT
106  REM  DEL LINE 470 IF YOU ARE      USING A SERIAL CARD
107  REM  FUNCTION IN LINE 190 **
108  :
120  DIM X(91),Y(90),G$(36):DR = .06981317
130  N = 90
140  HOME : INPUT "SCALE LIMITS ?";Z
150  INPUT "GRAPH TITLE ?";G$
160  HOME : VTAB (11): NORMAL : PRINT  TAB( 9): FLASH : PRINT "CONVERTING
     DATA POINTS": NORMAL
170  FOR I = 1 TO N:D = DR * I
175  :
176  :
180  REM   ***** FUNCTION *****
190  F =  SIN (3 * D): REM  SAMPLE
196  :
200  X(I) =  INT (((F *  COS (D) / Z + 1) * 30) + .5)
210  Y(I) =  INT ((( - F *  SIN (D) / Z + 1) * 18) + .5)
220  NEXT
230  HOME : VTAB (12): PRINT  TAB( 11): FLASH : PRINT "SORTING DATA POINT
     S": NORMAL
235  CHECK = 0
236  NN = N - 1
237  IF NN = 0 THEN 320
240  FOR I = 1 TO NN:J = I + 1
260  IF Y(I) <  = Y(J) THEN 300
270  A = Y(I):B = X(I)
280  Y(I) = Y(J):X(I) = X(J)
290  Y(J) = A:X(J) = B
295  CHECK = 1
300  NEXT I
320  G$(18) = "----------------------------I----------------------------
     "
330  T1$ = "                            I                            "
340  NORMAL : HOME : VTAB (12): PRINT  TAB( 15): FLASH : PRINT "FORMATTIN
     G": NORMAL
350  FOR I = 1 TO 17
360  G$(I) = T1$
370  NEXT I
380  FOR I = 19 TO 34
390  G$(I) = T1$
400  NEXT I
410  FOR I = 1 TO N - 1
420  IF X(I) > 58 OR Y(I) > 34 THEN 450
430  IF X(I) < 1 OR Y(I) < 1 THEN 450
440  G$(Y(I)) =  LEFT$ (G$(Y(I)),X(I) - 1) + "*" +  RIGHT$ (G$(Y(I)),59 -
     X(I))
450  NEXT I
455  VTAB (12): PRINT "          PRINTING....."
460  PRINT  CHR$ (4);"PR#";SL
470  PRINT  CHR$ (9);"80N"
480  PRINT G$
490  PRINT  TAB( 27);"+Y = ";Z
```

Fig. 2-24. Listing for POLARPLT program.

Continued on next page.


```
500  PRINT " "
510  FOR I = 1 TO 17
520  PRINT G$(I)
530  NEXT I
540  PRINT G$(18);"  +X = ";Z
550  FOR I = 19 TO 35
560  PRINT G$(I)
570  NEXT I
580  PRINT " "
590  PRINT  TAB( 27);"-Y = ";Z
600  FOR I = 1 TO 27: PRINT " ": NEXT I
610  PRINT  CHR$ (4);"PR#0"
615  PRINT  CHR$ (4);"PR#0"
620  HOME
625  VTAB (12): PRINT "        COMPLETED ....      "
630  END
```

Fig. 2-24 (cont). Listing for POLARPLT program.

```
        N-LEAVED ROSE  F=SIN(3*D)
                                          +Y = 1.2

                                            I
                                            I
                                            I
```

3

The Analysis of Signals

This chapter discusses programs which are used to analyze both periodic and aperiodic (nonrepetitive) waveforms. In particular, we will be able to determine both the average and root mean square (rms) values of a periodic function, the representation of any periodic signal by Fourier series, the representation of aperiodic signals by a modification of the Fourier transform, and analyze damped oscillations.

THE AVERAGE AND RMS VALUES OF A WAVEFORM

The RMSAV program listing shown in Fig. 3-1 determines both the average and rms values of a periodic signal. This is extremely useful in verifying ac measurements, such as determining the average power of a circuit, or of a particular device within the circuit. Mathematically, the average value of a periodic waveform f(t) is:

$$AV = \frac{1}{T}\int_0^T f(t)\ dt \qquad \text{(Eq. 3-1)}$$

However, it must be noted that the average value for most periodic functions having symmetry about zero amplitude (i.e., no dc term) is naturally zero, since the same amount of the waveform is above as well as below the zero reference, like most sine, triangular, and square waves. When this occurs, a half-cycle average value is determined to represent the average value of a periodic function, so that

$$AV = \frac{2}{T}\int_0^{T/2} f(t)\ dt \qquad \text{(Eq. 3-2)}$$

On the other hand, the rms value is given by:

$$RMS = \left(\frac{1}{T}\int_0^T f(t)^2\ dt\right)^{1/2} \qquad \text{(Eq. 3-3)}$$

Using Simpson's rule to numerically compute the integrals, the RMSAV program determines the average and rms values from equally spaced discrete data points, or from a periodic function, which must be supplied in line 124, so as the argument of the periodic function must have the form:

argument = (A∗I)/57.92577

to convert the angle from degrees to radians. If the average value is zero, then the half-cycle average is determined. Furthermore, Simpson's rule requires that the number of data points be an odd number. In addition, careful attention must be paid to discontinuities which occur during the cycle, as will be illustrated in a later example.

Example 3-1

Determine the average and rms values of an ordinary 60-Hz sine wave whose amplitude varies between −1.0 and +1.0 volts. Fig. 3-2 shows the output for the RMSAV program, when the sine wave function is supplied in line 124, giving 51 data points for one complete cycle. Since the full cycle average value for this sine wave is zero, the program then computes the half-cycle average in addition to the rms value. Using classical methods, the half-cycle average and rms values for a sine wave are 0.637 and 0.707 times the peak voltage.

Example 3-2

Consider the 50% duty cycle square wave, shown in Fig. 3-3, which varies between 0 and 5 volts. We will first run the RMSAV program by entering discrete data points instead of describing

```
100  REM  RMS AND AVG VALUE OF A PERIODIC WAVEFORM (RMSAV)
101  CALL  - 936: VTAB (5): PRINT  TAB( 8): INVERSE : PRINT "RMS AND AVER
     AGE VALUE OF": NORMAL
102  VTAB (7): PRINT  TAB( 11): INVERSE : PRINT "A PERIODIC WAVEFORM": NORMAL
103  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980"
104  PRINT : PRINT  TAB( 9)"HOWARD M. BERLIN, LTD."
105  PRINT : PRINT  TAB( 16)"APPLE II"
106  PRINT : PRINT  TAB( 11)"APPLESOFT VERSION"
107  FOR I = 1 TO 2500: NEXT I: CALL  - 936
108  CLEAR : DIM C(361):RAD = 57.29578
109  PRINT : PRINT "RMS AND AVERAGE VALUES ARE TO BE": PRINT "DETERMINED
     FROM:": PRINT : PRINT  TAB( 7)"1. DISCRETE POINTS": PRINT : PRINT  TAB(
     7)"2. A DEFINED FUNCTION"
110  PRINT : PRINT : PRINT "FUNCTION MUST BE DEFINED IN LINE 124.": PRINT
     : PRINT  TAB( 14)"FOR EXAMPLE:": PRINT : PRINT : PRINT "124 C(I)=SIN
     ((A*I)/57.29577)"
111  PRINT : PRINT : INPUT "SELECT CHOICE? ";Z9
112  PRINT : INPUT "CYCLE PERIOD? ";T
113  PRINT : PRINT "NUMBER OF DIVISIONS FOR 1 CYCLE": INPUT "  (MUST BE A
     N INTEGER)  -->  ? ";C
114  V = T / C: IF Z9 <  > 1 THEN 122
116  CALL  - 936: PRINT "TIME"
117  FOR I = 0 TO C
118  PRINT I * V;" ";
119  INPUT C(I)
120  NEXT I
121  CALL  - 936: GOTO 126
122  A = 360 / C: HOME : GOSUB 150
123  FOR I = 0 TO C
124  C(I) =  SIN ((A * I) / RAD)
125  NEXT I
126  Z = 1 / T
127  S = 0:AV = 0:HC = 0
129  FOR I = 0 TO C - 2 STEP 2
130  S = S + C(I) ^ 2 + 4 * C(I + 1) ^ 2 + C(I + 2) ^ 2
131  AV = AV + C(I) + 4 * C(I + 1) + C(I + 2)
132  IF I <  = C / 2 THEN  GOSUB 143
133  NEXT I
134  AV = Z * V * AV / 3: IF AV < 1E - 3 THEN AV = 0
135  HC = Z * V * HC * 2 / 3
136  RS =  SQR (Z * V * S / 3)
137  CALL  - 936: VTAB 8: PRINT "RMS VALUE = ";RS
138  IF AV = 0 THEN 141
139  PRINT : PRINT "AVERAGE VALUE = ";AV
140  IF AV <  > 0 THEN 142
141  PRINT : PRINT "HALF CYCLE AVG. VALUE = ";HC
142  END
143  HC = HC + C(I) + 4 * C(I + 1) + C(I + 2): RETURN
150  CALL  - 936: VTAB (10): PRINT  TAB( 15): FLASH : PRINT "COMPUTING": NORMAL : RETURN
```

Fig. 3-1. Listing for RMSAV program.

the function. Square waves are unusual functions in that they have discontinuities. In order to minimize computational error, it is best to assign a value that is midway between the minimum and maximum values of the square wave (i.e., 2.5 volts) as the value of the point where the discontinuity occurs during the cycle. For the situation where the square wave is discontinuous at both the beginning and the end of the cycle, we use the values of 5 and 0 volts respectively. When numerical integration methods are used, greater accuracy is achieved when the number of points is made as large as possible. Fig. 3-4 shows the results when 21 data points are used (20 cycle divisions plus the value at t = 0).

Using classical methods to check our results,


```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 2

CYCLE PERIOD? 0.16667

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?50

RMS VALUE = .707106723

HALF CYCLE AVERAGE VALUE = .634110557
```

Fig. 3-2. Output for RMSAV program.

we would find that the average value of such a waveform is one-half its peak value, or 2.5, while the rms value is 0.707 times the peak, or 3.535. Increasing the number of data points usually leads to greater accuracy.

However, suppose we had described the square wave mathematically instead of by discrete points. Unfortunately, we still have difficulty in handling the discontinuity at t = 0.5 second. One way around this problem is to run the program twice and then average the results. In the original RMSAV listing, lines 123–125 are replaced by the following statements:

```
123 FOR I=0 TO (C/2)-1: C(I)=5: NEXT I
124 C(C/2)=5
125 FOR I=(C/2)+1 TO C: C(I)=0: NEXT I
```

so that the results of the first run will be slightly higher than the expected value, as shown in Fig. 3-5A. Then replace line 124 by C(C/2) = 0 so that the results of the second run will be slightly lower than the expected values, as shown in Fig. 3-5B. The two results are then averaged, in which case the average value is the average of 2.51666667 and 2.48333333, or 2.5, while the rms value is

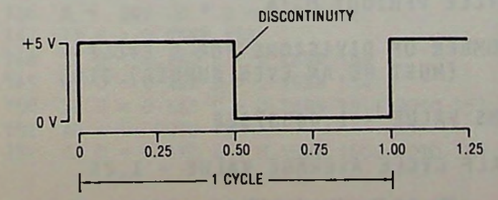


Fig. 3-3. 50% duty cycle square wave.

```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 1

CYCLE PERIOD? 1.0

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?20

TIME
0 ?5
.05 ?5
.1 ?5
.15 ?5
.2 ?5
.25 ?5
.3 ?5
.35 ?5
.4 ?5
.45 ?5
.5 ?2.5
.55 ?0
.6 ?0
.65 ?0
.7 ?0
.75 ?0
.8 ?0
.85 ?0
.9 ?0
.95 ?0
1 ?0

RMS VALUE = 3.50594733

AVERAGE VALUE = 2.5
```

Fig. 3-4. Results when 21 data points are used.

the average of 3.54729944 and 3.52372909, or 3.535514265.

Example 3-3

Determine the average and rms values for the sawtooth waveform shown in Fig. 3-6. Unlike the square wave which has a discontinuity in the middle as well as at the beginning and end of the cycle, the sawtooth is discontinuous only at the beginning and end of the cycle.

```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 2

CYCLE PERIOD? 1.0

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?100

RMS VALUE = 3.54729944

HALF CYCLE AVERAGE VALUE = 2.51666667
```

(A) Overestimate.

```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 2

CYCLE PERIOD? 1.0

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?100

RMS VALUE = 3.52372909

HALF CYCLE AVERAGE VALUE = 2.48333333
```

(B) Underestimate.

Fig. 3-5. Rms and average values of a periodic waveform.

Fig. 3-7 shows the results using 11 data points, with no corrections made for the pair of discontinuities. The starting point is zero, while the last

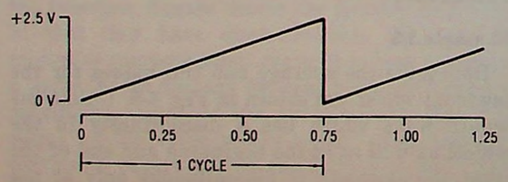


Fig. 3-6. Sawtooth waveform.


```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 1

CYCLE PERIOD? 0.75

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?10

TIME
0 ?5
.075 ?.25
.15 ?.5
.225 ?.75
.3 ? 1.0
.375 ?1.25
.45 ?1.5
.525 ?1.75
.6 ?2.0
.675 ?2.25
.75 ?2.5

RMS VALUE = 1.44337567

AVERAGE VALUE = 1.25
```

Fig. 3-7. Results when 11 data points are used.

```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 2

CYCLE PERIOD? 0.75

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?100

RMS VALUE = 1.44337568

HALF CYCLE AVERAGE VALUE = 1.25
```

Fig. 3-8. Results when 100 data points are used.

```
100  REM  FOURIER SERIES PROGRAM (FSERIES)
101  HOME : VTAB (7): PRINT  TAB( 13): INVERSE : PRINT "FOURIER SERIES": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II: HOME
106  PRINT : PRINT "FOURIER SERIES TO BE COMPUTED FROM:"
107  PRINT : PRINT "     1. DISCRETE DATA POINTS"
108  PRINT : PRINT "     2. AN EQUATION"
109  PRINT : PRINT : PRINT "EQUATION MUST BE DEFINED IN LINE 168": PRINT
     : PRINT "FOR EXAMPLE:"
110  PRINT : PRINT "168 F(Y) = 162 * COS (6.28 * I)"
111  PRINT : PRINT : INPUT "SELECT? ";Z9
112  PRINT : DIM F(100),C(100),D(100),E(100),A2(100)
113  IF Z9 <  > 1 THEN  PRINT : PRINT "COMPUTING ...": GOTO 167
114  HOME : PRINT : INPUT "NUMBER OF DATA POINTS? ";N
115  IF (N / 2) =  INT ((N / 2) + .5) THEN 114
116  PRINT : INPUT "NUMBER OF HARMONICS? ";N8: PRINT
117  IF N8 > (N / 2) - 2 THEN 116
118  IF Z9 <  > 1 THEN 123
119  FOR I = 1 TO N
120  PRINT "POINT #";I;"?";
121  INPUT " ";F(I)
122  NEXT I
123  PRINT : INPUT "SIGNAL PERIOD? ";T
124  N3 = N - 1
125  A = 2 * 3.1415927 / T
126  T3 = T / N3
127  A0 = 0
128  FOR I = 1 TO N3 STEP 2
129  A0 = A0 + F(I) + 4 * F(I + 1) + F(I + 2)
130  NEXT I
131  A0 = A0 * T3 / (3 * T)
132  IF  ABS (A0) < .001 THEN A0 = 0
133  HOME : VTAB (9): PRINT  TAB( 8): FLASH : PRINT "COMPUTATIONS IN PROG
     RESS": NORMAL
134  PRINT : FOR X = 1 TO N8
135  HTAB 19: PRINT X;
136  C = 0:D = 0:E = 0:A2 = 0:V = X * A
137  T1 =  - T3
138  FOR J = 1 TO N3 STEP 2
139  T1 = T1 + 2 * T3
140  C = C + (F(J) *  COS (V * (T1 - T3))) + (4 * F(J + 1) *  COS (V * T1)
     ) + (F(J + 2) *  COS (V * (T1 + T3)))
141  D = D + (F(J) *  SIN (V * (T1 - T3))) + (4 * F(J + 1) *  SIN (V * T1)
     ) + (F(J + 2) *  SIN (V * (T1 + T3)))
142  NEXT J
143  C = 2 * C * T3 / (3 * T):D = 2 * D * T3 / (3 * T)
144  IF  ABS (C) < .005 THEN C = 0
145  IF  ABS (D) < .005 THEN D = 0
146  E =  SQR (C * C + D * D)
147  IF C > 0 THEN 155
148  IF C < 0 AND D < 0 THEN 156
149  IF C = 0 AND D > 0 THEN 154
150  IF C = 0 AND D = 0 THEN 151: GOTO 152
151  A2 = 0: GOTO 158
152  IF C = 0 AND D < 0 THEN 153: GOTO 157
```

Continued on next page.

Fig. 3-9. Listing for FSERIES program.

```
153  A2 =  - 90: GOTO 158
154  A2 = 90: GOTO 158
155  A2 = ( ATN (D / C)) * 180 / 3.14159: GOTO 158
156  A2 =  - 180 + ( ATN (D / C)) * 180 / 3.14159: GOTO 158
157  A2 = 180 + ( ATN (D / C)) * 180 / 3.14159
158  C(X) = C:D(X) = D:E(X) = E:A2(X) = A2
159  D(X) =  INT (D(X) * 1E4 + .5) / 1E4:E(X) =  INT (E(X) * 1E4 + .5) / 1
     E4:C(X) =  INT (C(X) * 1E4 + .5) / 1E4:A2(X) =  INT (A2(X) * 100 + .
     5) / 100: NEXT X
160  HOME : PRINT : PRINT "DC TERM, A(0) = ";A0
161  PRINT : PRINT  TAB( 11)"COS"; TAB( 17)"SIN": PRINT "HARMONIC  TERM
     TERM  MAGNITUDE PHASE"
162  FOR I = 1 TO N8
163  PRINT  TAB( 4)I; TAB( 11)C(I); TAB( 17)D(I); TAB( 26)E(I); TAB( 34)A
     2(I): NEXT I
164  F = 1 / T:F =  INT (F * 1E4 + .5) / 1E4
165  PRINT : PRINT "FUNDAMENTAL FREQUENCY = ";F;" HZ"
166  PRINT : END : REM
167  Y = 1: FOR I = 0 TO 1 STEP .025: REM  SAMPLE FUNCTION
168  F(Y) = 162 *  COS (6.283185 * I): REM  SAMPLE FUNCTION
169  Y = Y + 1: NEXT I:N = Y - 1: REM  SAMPLE FUNCTION
170  CALL  - 998: CALL  - 998: GOTO 116: REM CURSOR UP & RETURN
```

Fig. 3-9 (cont). Listing for FSERIES program.

point is the peak value of +2.5 volts. Using classical methods, the sawtooth has an average value that is one-half its peak, or 1.25 volts, while the rms value is 0.577 times the peak, or 1.443 volts.

Like the square wave, we can repeat this example by defining the waveform mathematically as a straight line with a slope equal to (2.5/0.75). Therefore, lines 123–125 would read as:

```
123 FOR I=0 TO C
124 C(I)=2.5*I/C
125 NEXT I
```

By dividing the waveform into 100 increments, the output results of Fig. 3-8 show that there is no difference!

FOURIER SERIES

Any periodic waveform can be expressed in terms of a Fourier series provided that it has a finite average value over one complete cycle; there are a finite number of discontinuities, if any exist, over one complete cycle; and it has a finite number of maximum positive and negative values. When these three conditions are met, a Fourier series for a periodic waveform f(t) exists, and can be written by the following trigonometric form:

$$f(t) = A_0 + A_1\cos(\omega t) + A_2\cos(2\omega t) + \ldots + B_1\sin(\omega t) + B_2\sin(2\omega t) + \ldots \quad \text{(Eq. 3-4)}$$

consisting of a dc, or constant term, and an infinite series of harmonically related sines and cosines. However, Equation 3-4 may be stated in a form using only cosine and phase angle terms, so that:

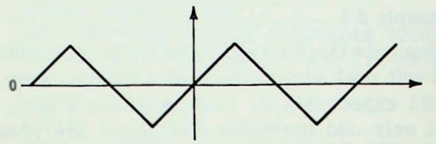
$$f(t) = C_0 + C_1\cos(\omega t + \phi_1) + C_2\cos(2\omega t + \phi_2) + \ldots \quad \text{(Eq. 3-5)}$$

Equation 3-5 is sometimes referred to as the magnitude and phase Fourier series, and is related to Equation 3-4 by:

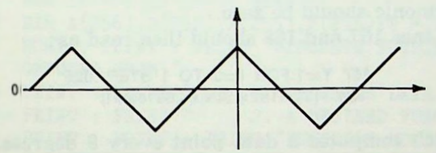
$$A_n = C_n\cos(n\omega t) \quad \text{(Eq. 3-6A)}$$
$$B_n = C_n\sin(n\omega t) \quad \text{(Eq. 3-6B)}$$
$$C_n = (A_n^2 + B_n^2)^{1/2} \quad \text{(Eq. 3-6C)}$$
$$\phi_n = \tan^{-1}(B_n/A_n) \quad \text{(Eq. 3-6D)}$$

The FSERIES program listed in Fig. 3-9 calculates the A, B, and C coefficients in addition to the phase angle ϕ for a given periodic waveform. As was the case with the RMSAV program, the waveform may be expressed either as a number of equally spaced data points, or as a periodic function of time, which must be supplied in line 168, which assumes the form that 1 complete cycle equals 6.283185 radians. Line 167 sets the sampling rate as a fraction of 1 cycle. For the values shown, the function in line 168 will be computed every 1/40th of a cycle (i.e., 1/.025), or every 9°.

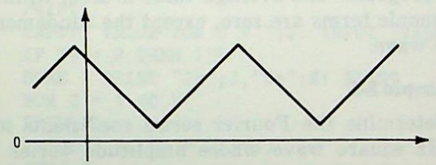
If equally spaced discrete data points are used to describe the waveform, the number of data points must be an odd number, while the number of coefficients must be less than half the number of data points minus 2. Like the RMSAV program, careful attention must be paid to functions that have discontinuities, whose number must be finite.



(A) Odd function with zero average value.



(B) Even function with zero average value.



(C) Nonzero average value.

Fig. 3-10. Symmetry functions.

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DISCRETE DATA POINTS

     2. AN EQUATION

EQUATION MUST BE DEFINED IN LINE 168
FOR EXAMPLE:

168 F(Y) = 162 * SIN (6.28 * I)

SELECT? 2

NUMBER OF HARMONICS? 7

SIGNAL PERIOD? 0.01667

DC TERM, A(0) = 0

            COS     SIN
HARMONIC    TERM    TERM    MAGNITUDE  PHASE
   1          0     162        162       0
   2          0       0          0       0
   3          0       0          0       0
   4          0       0          0       0
   5          0       0          0       0
   6          0       0          0       0
   7          0       0          0       0

FUNDAMENTAL FREQUENCY = 59.988 HZ
```

Fig. 3-11. Fourier series computed.

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DISCRETE DATA POINTS

     2. AN EQUATION

EQUATION MUST BE DEFINED IN LINE 168
FOR EXAMPLE:

168 F(Y) = 162 * SIN (6.28 * I)

SELECT? 2

NUMBER OF DATA POINTS? 21

NUMBER OF HARMONICS? 7

POINT #1? 1
POINT #2? 1
POINT #3? 1
POINT #4? 1
POINT #5? 1
POINT #6? 1
POINT #7? 1
POINT #8? 1
POINT #9? 1
POINT #10? 1
POINT #11? 0
POINT #12? -1
POINT #13? -1
POINT #14? -1
POINT #15? -1
POINT #16? -1
POINT #17? -1
POINT #18? -1
POINT #19? -1
POINT #20? -1
POINT #21? -1

SIGNAL PERIOD? 0.1

DC TERM, A(0) = 0

            COS     SIN
HARMONIC    TERM    TERM    MAGNITUDE  PHASE
   1          0     1.2733    1.2733    90
   2          0       0          0       0
   3          0     .4265     .4265     90
   4          0       0          0       0
   5          0     .2667     .2667     90
   6          0       0          0       0
   7          0     .2327     .2327     90

FUNDAMENTAL FREQUENCY = 10 HZ
```

Fig. 3-12. Results when 21 data points are used.

When the coefficients are evaluated, there may be some slight errors, especially since all computations are done with single precision arithmetic. However, by recognizing the symmetrical behavior of the particular function, one can predict which coefficients can be ignored.

Waveforms with certain types of symmetry are void of particular harmonics, such that:

1. Odd functions have only sine terms, so that all cosine coefficients are zero.
2. Even functions have only cosine terms, so that all sine terms are zero.
3. Functions that exhibit half-wave symmetry have only odd harmonics present.
4. Functions that are symmetrical about the independent variable axis have an average value (i.e., the dc or constant term) equal to zero.

Using triangle waves, Fig. 3-10 illustrates examples of those which are odd or even functions, those having only odd harmonics, and those having zero average value.

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DISCRETE DATA POINTS

     2. AN EQUATION

EQUATION MUST BE DEFINED IN LINE 168
FOR EXAMPLE:

168 F(Y) = 162 * SIN (6.28 * I)

SELECT? 2

NUMBER OF HARMONICS? 7

SIGNAL PERIOD? 0.75


DC TERM, A(0) = 1.25

          COS    SIN
HARMONIC  TERM   TERM   MAGNITUDE PHASE
   1      0     -.7958   .7958   -90
   2      0     -.3980   .3980   -90
   3      0     -.2655   .2655   -90
   4      0     -.1995   .1995   -90
   5      0     -.1604   .1604   -90
   6      0     -.1349   .1349   -90
   7      0     -.1176   .1176   -90

FUNDAMENTAL FREQUENCY = 1.3333 HZ
```

Fig. 3-13. Results with all cosine terms zero.

Example 3-4

Compute the Fourier series for a simple 60-Hz, 162-volt peak sine wave. From its symmetry, we would expect that its average value is zero, and that only odd harmonic sine terms are present. Furthermore, since we have a pure sine wave, all harmonics except the fundamental, or first harmonic should be zero.

Lines 167 and 168 should then read as:

```
167 Y=1:FOR I=0 TO 1 STEP .025
168 F(Y)=162*SIN(6.283185*I)
```

which computes a data point every 9 degrees, or 1/40th of a cycle. Fig. 3-11 shows the results of the program. The average value is zero, while all harmonic terms are zero, except the fundamental sine wave.

Example 3-5

Determine the Fourier series coefficients of a 10-Hz square wave whose amplitude varies between −1 and +1 volts. Since such a wave is both an odd function and has half-wave symmetry, we would then expect that the average value, all cosine terms, and all even sine terms, will be zero. Fig. 3-12 shows the output results when 21 data points are used, so that:

Points 1–10: +1
Point 11: 0 (the half-cycle discontinuity)
Points 12–21: −1

As a comparison, it can be shown that the Fourier series for this square wave is:

$$f(t) = \frac{4V}{\pi} \sum_{n=0}^{m} \frac{\sin(2n+1)\omega t}{2n+1} \quad \text{(Eq. 3-7)}$$

If desired, the square wave can be expressed mathematically as was the case for determining the average and rms values in Example 3-2, and averaging the results of two runs. This is left as an optional exercise for the reader.

Example 3-6

Determine the Fourier coefficients for the sawtooth wave previously shown in Fig. 3-6. From its symmetry, we would expect that, since it is an odd function, only sine terms will be present.

Fig. 3-13 shows the results, in which all cosine terms are zero. As a comparison, it can be shown that the Fourier series for this sawtooth is:

$$f(t) = \frac{V}{2} - \frac{V}{\pi} \sum_{n=1}^{m} \frac{\sin(n\omega t)}{n} \quad \text{(Eq. 3-8)}$$


```
100  LOMEM: 16384: CLEAR
105  REM  DISCRETE FOURIER TRANSFORM PROGRAM (DFTRANS)
110  HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "DISCRETE FOURIER T
     RANSFORM": NORMAL
115  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
120  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 12)"APPLESOFT
     VERSION"
125  FOR II = 1 TO 2500: NEXT II
130  DIM A(256),B(256),X(256),Y(256):T = 1000:V = 8
135  HOME : PRINT : PRINT "DISCRETE FOURIER TRANSFORM": PRINT "IS TO BE C
     OMPUTED FROM:"
140  PRINT : PRINT "     1. DISCRETE DATA POINTS"
145  PRINT : PRINT "     2. A DEFINED FUNCTION"
150  PRINT : PRINT : PRINT "FUNCTION MUST BE DEFINED": PRINT "ACCORDING T
     O INSTRUCTIONS ": FLASH : PRINT "IN PAMPHLET": NORMAL
155  PRINT : INPUT "SELECT? ";Z9
160  PRINT
165  INPUT "VALUE FOR J ? ";J: INPUT "VALUE FOR N ? ";N
170  IF Z9 = 2 THEN 195
172  HOME : PRINT "J=";J,"N=";N: PRINT
175  FOR Z = 1 TO N
180  PRINT Z;
185  INPUT " VALUE = ";A(Z)
190  NEXT Z
195  P = 3.1415926535
200  HOME : VTAB (7): PRINT  TAB( 5)"CALCULATION OF INPUT FUNCTION": PRINT
     : PRINT  TAB( 14): FLASH : PRINT "IN PROGRESS": NORMAL
205  IF Z9 = 1 THEN 240
210  FOR Z = 0 TO N * 7 / 8
215  A(Z) = 1
220  NEXT Z
225  FOR Z = N * 7 / 8 TO N
230  A(Z) = 0
235  NEXT Z
240  FOR Z = 0 TO N - 1
245  A(Z) = A(Z) / N
250  NEXT Z
255  C = N / 2:D = 1:E = P * 2 / N: PRINT
260  FOR I = 1 TO J: HTAB (19): PRINT I: CALL  - 998
265  F = 0:G = C
270  FOR S = 1 TO D
275  H =  INT (F / C)
280  GOSUB 465
285  Q = R
290  A1 =  COS (E * Q):A2 =  -  SIN (E * Q)
295  FOR K = F TO G - 1
300  A3 = A(K):A4 = B(K)
305  B1 = A1 * A(K + C) - A2 * B(K + C)
310  B2 = A2 * A(K + C) + A1 * B(K + C)
315  A(K) = A3 + B1:B(K) = A4 + B2
320  A(K + C) = A3 - B1:B(K + C) = A4 - B2
325  NEXT K
330  F = F + 2 * C:G = G + 2 * C
335  NEXT S
340  C = C / 2:D = D * 2
345  NEXT I
350  GOTO 355
355  U = 0
```

Continued on next page.

Fig. 3-14. Listing for DFTRANS program.

```
360  Z = 0
365  HOME
370  PRINT "HM"; TAB( 5)"RE(Z)"; TAB( 19)"IM(Z)"; TAB( 33)"Z"
375  PRINT "==";  TAB( 5)"======"; TAB( 19)"======"; TAB( 32)"======"
380  H = U
385  GOSUB 505
390  V1$ =  LEFT$ (( STR$ (U)),V)
391  V2 = ( INT ((A(R)) * 10 ^ 7 + .5)) / 10 ^ 7
392  V3 = ( INT ((B(R)) * 10 ^ 7 + .5)) / 10 ^ 7
393  V4 = ( INT (ZZ * 10 ^ 7 + .5)) / 10 ^ 7
395  PRINT  TAB( 1)V1$; TAB( 4)V2; TAB( 16)V3; TAB( 28)V4
400  Y(U) = ZZ:X(U) = U
405  U = U + 1:Z = Z + 1
410  IF Z <  = 13 THEN 430
415  PRINT : PRINT "HIT RETURN TO CONTINUE";:
420  POKE  - 16368,0: CALL  - 756: HOME
425  GOTO 360
430  IF U > N / 2 THEN 440
435  GOTO 380
440  PRINT : PRINT "HIT RETURN FOR SPECTRUM PLOT"
445  POKE  - 16368,0:N = U: GOTO 530
455  GOTO 530
465  R = 0:N1 = N
470  FOR W = 1 TO J
475  N1 = N1 / 2
480  IF H < N1 THEN 495
485  R = R + 2 ^ (W - 1)
490  H = H - N1
495  NEXT W
500  RETURN
505  GOSUB 465
510  ZZ =  SQR (A(R) ^ 2 + B(R) ^ 2)
515  RETURN
520  END
525  REM    SPECTRUM PLOT ROUTINE
530  Y1 = Y(0):Y2 = Y(0):X1 = X(0):X2 = X(0)
535  FOR I = 1 TO N
540  IF (Y1 - Y(I)) <  = 0 THEN 555
545  Y1 = Y(I)
550  GOTO 565
555  IF (Y2 - Y(I)) >  = 0 THEN 565
560  Y2 = Y(I)
565  IF (X(I)) <  = 0 THEN 580
570  X1 = X(I)
575  GOTO 590
580  IF (X2 - X(I)) >  = 0 THEN 590
585  X2 = X(I) + 1
590  : NEXT I: CALL  - 756
595  XD = X1 - X2:YD = Y2 - Y1
600  CALL 62450: HGR : HCOLOR= 7
605  HPLOT 0,145 TO 278,145
610  FOR X = 0 TO 278 STEP 278 / 8: HPLOT X,150 TO X,145: NEXT X
615  FOR I = 0 TO N - 1
620  X = 278 - ((X1 - X(I)) * 278 / XD):Y = 141 - ((Y(I) - Y1) * 141 / YD)
625  IF X > 279 THEN 635
630  HPLOT X,Y TO X,141
635  NEXT I
```

Continued on next page.

Fig. 3-14 (cont). Listing for DFTRANS program.


```
640  TB = 39: IF X1 > 99 THEN TB = 38
645  VTAB (21): PRINT "0"; TAB( 14)"HARMONIC NUMBER"; TAB( TB)X1
650  POKE  - 16368,0: CALL  - 756
655  HOME : TEXT : VTAB (12): PRINT "GOODBYE ........"
660  END
```

Fig. 3-14 (cont). Listing for DFTRANS program.

FOURIER TRANSFORM

Whereas the Fourier series is used to represent the harmonic, or spectral content, of any periodic function, the Fourier transform is intended for those signals which are not periodic, but nevertheless continuous. The mathematical transformation of a function of time f(t) into a function of frequency F(s) is given by the equation:

$$F(s) = \int_{-\infty}^{\infty} f(t)e^{-st}\,dt \qquad \text{(Eq. 3-9)}$$

Instead of being able to represent the Fourier transform of a continuous function as a continuous function of frequency, the DFTRANS program of Fig. 3-14 computes a discrete Fourier

HARM	RE(Z)	IM(Z)	Z
0	.875	0	.875
1	-.11016	-.0521	.12186
2	-.07151	-.07713	.11272
3	-.0239	-.09543	.09839
4	.01562	-.07855	.08009
5	.03539	-.04772	.05941
6	.03357	-.01794	.03806
7	.1373	-8.7E-04	.01375
8	0	0	0
9	-9.39E-03	-.01036	.01399

Fig. 3-15. Output results for first nine harmonics.

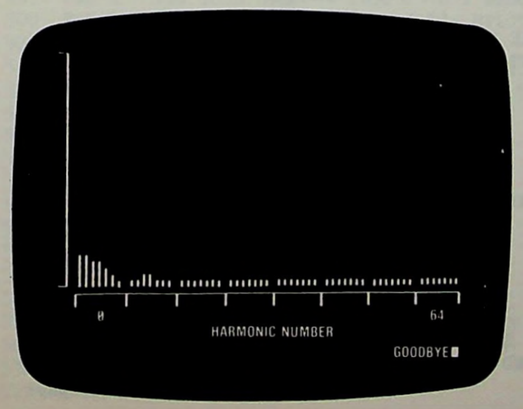


Fig. 3-16. Spectrum plot.

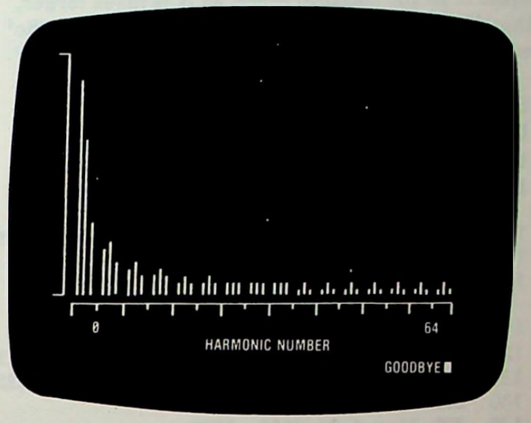


Fig. 3-17. Spectrum plot for a 25% duty cycle pulse.

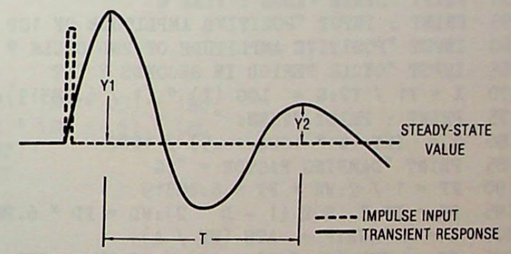


Fig. 3-18. Transient response to an impulse input.

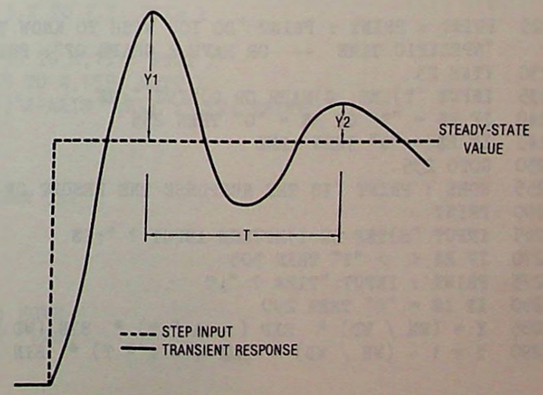


Fig. 3-19. Transient response to a step input.

transform so that the frequency function is now approximated by discrete points instead of a continuous function.

The continuous time function is sampled N times, so that $N = 2^J$, where J is an integer. If the number of sample points is 64, then J must equal 6. In the classical sense, as N is made larger, the discrete Fourier transform approaches the continuous function described by Equation 3-9.

When run, the DFTRANS program requires that the number of data point samples (N) in line 165 be as shown in Table 3-1; consequently, the value for J must also be defined as summarized in Table 3-1. Because this program generates a spectrum plot that can contain up to 128 discrete frequencies, the maximum number of sampled data points is 256, in which case J = 8.

Table 3-1. DFTRANS Program

Data Points		Discrete Frequencies
N	J	
16	4	8
32	5	16
64	6	32
128	7	64
256	8	128

```
100  REM  ANALYSIS OF DAMPED OSCILLATIONS (DAMPED)
105  LOMEM: 16385
110  HOME : VTAB 6: PRINT  TAB( 6): INVERSE : PRINT "ANALYSIS OF DAMPED O
     SCILLATIONS": NORMAL
115  VTAB 20
120  PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
125  VTAB (13): PRINT  SPC( 12)"APPLESOFT VERSION"
130  DIM X(267),Y(267)
135  FOR II = 1 TO 2500: NEXT
140  HOME
145  PRINT "AMPLITUDES ARE MEASURED FROM THE STEADY"
150  PRINT "STATE VALUE": VTAB 6
155  PRINT : INPUT "POSITIVE AMPLITUDE OF 1ST CYCLE ? ";Y1
160  INPUT "POSITIVE AMPLITUDE OF 2ND CYCLE ? ";Y2
165  INPUT "CYCLE PERIOD IN SECONDS ? ";T
170  X = Y1 / Y2:G =  LOG (X) * (1 / 6.28319):D = G /  SQR (1 + G ^ 2)
175  PRINT : PRINT "THEN: "
180  G =  INT (G * 1000 + .5) / 1000
185  PRINT "DAMPING FACTOR = ";G
190  FF = 1 / T:WN = FF * 6.28319
195  FD = FF *  SQR (1 - D ^ 2):WD = FD * 6.28319
200  A = D * WN:P =  ATN (WN / A)
205  FF =  INT (FF * 10 + .5) / 10:WN =  INT (WN * 10 + .5) / 10
210  FD =  INT (FD * 10 + .5) / 10:WD =  INT (WD * 10 + .5) / 10
215  PRINT "NATURAL FREQUENCY = ";FF;" HZ": PRINT "   (";WN;" RAD/SEC)"
220  PRINT "DAMPED NATURAL FREQ  = ";FD;" HZ ": PRINT "   (";WD;" RAD/SEC
     )"
225  PRINT : PRINT : PRINT "DO YOU WISH TO KNOW THE RESPONSE AT A": PRINT
     "SPECIFIC TIME  --  OR HAVE A GRAPH OF": PRINT "THE WAVEFORM  ?"
230  VTAB 23
235  INPUT "T)IME, G)RAPH OR Q)UIT? ";R$
240  IF R$ = "T" OR R$ = "G" THEN 255
245  IF R$ = "Q" THEN  END
250  GOTO 235
255  HOME : PRINT "IS THE RESPONSE THE RESULT OF A:"
260  PRINT
265  INPUT "S)TEP OR I)MPULSE INPUT ? ";I$
270  IF R$ <  > "T" THEN 305
275  PRINT : INPUT "TIME ? ";T
280  IF I$ = "S" THEN 290
285  Y = (WN / WD) *  EXP ( - A * T) *  SIN (WD * T): GOTO 295
290  Y = 1 - (WN / WD) *  EXP ( - A * T) *  SIN (WD * T + P)
```

Continued on next page.

Fig. 3-20. Listing for DAMPED program.


```
295  PRINT "AMPLITUDE = ";Y
300  GOTO 225
305  HOME : INPUT "ENTER MIN,MAX TIME ";XA,XB
310  PRINT
315  N = 266:I = 1:C = (XB - XA) / 265
320  FOR XT = XA TO XB + C STEP C
325  IF I$ = "S" THEN 335
330  Y = (WN / WD) *  EXP ( - A * XT) *  SIN (WD * XT): GOTO 340
335  Y = 1 - (WN / WD) *  EXP ( - A * XT) *  SIN (WD * XT + P)
340  X(I) = XT:Y(I) = Y:I = I + 1: PRINT XT;"         ": CALL  - 998: NEXT
     XT
345  CALL  - 868: HTAB 1
350  GOSUB 365
355  PRINT : END
360  REM  X-Y PLOT SUBROUTINE
365  T$ = "STEP RESPONSE"
370  IF I$ = "I" THEN T$ = "IMPULSE RESPONSE"
375  REM  FIND MIN,MAX OF X & Y
380  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
385  FOR I = 2 TO N
390  IF (Y1 - Y(I)) <  = 0 THEN 405
395  Y1 = Y(I)
400  GOTO 415
405  IF (Y2 - Y(I)) >  = 0 THEN 415
410  Y2 = Y(I)
415  IF (X1 - X(I)) <  = 0 THEN 430
420  X1 = X(I)
425  GOTO 440
430  IF (X2 - X(I)) >  = 0 THEN 440
435  X2 = X(I): PRINT I: CALL  - 998
440  NEXT I
445  X1 =  INT (X1 * 1E6 + .5) / 1E6:X2 =  INT (X2 * 1E6 + .5) / 1E6
450  Y1 =  INT (Y1 * 1E6 + .5) / 1E6:Y2 =  INT (Y2 * 1E6 + .5) / 1E6
455  PRINT "MIN-MAX X VALUES: ": PRINT X1,X2: PRINT
460  PRINT "MIN-MAX Y VALUES: ": PRINT Y1,Y2: PRINT
465  INPUT "DESIRED MIN.MAX X-AXIS ? ";X1,X2
470  INPUT "DESIRED MIN,MAX Y-AXIS ? ";Y1,Y2
475  XD = X2 - X1:YD = Y2 - Y1
480  REM  ------ GRAPH ---------
485  GR : HGR : HCOLOR= 7: VTAB (23)
490  HPLOT 8,0 TO 8,150
495  FOR Y = 150 TO 0 STEP  - (150 / 10): HPLOT 6,Y TO 7,Y: NEXT Y
500  FOR Y = 150 TO 0 STEP  - (150 / 5): HPLOT 3,Y TO 7,Y: NEXT Y
505  HPLOT 13,154 TO 279,154
510  FOR X = 13 TO 279 STEP 266 / 10: HPLOT X,155 TO X,157: NEXT X
515  FOR X = 13 TO 279 STEP 266 / 5: HPLOT X,155 TO X,159: NEXT X
520  VTAB 24: PRINT "Y-AXIS ";Y1;":";Y2; SPC( 5)"X-AXIS ";X1;":";X2: PRINT
525  FOR I = 1 TO N
530  X = 279 - ((X2 - X(I)) * 266 / XD)
535  Y = 150 - ((Y(I) - Y1) * 150 / YD)
540  IF X > 279 OR X < 13 OR Y < 0 OR Y > 159 THEN 552
545  IF I = 1 THEN  HPLOT X,Y: GOTO 552
550  HPLOT  TO X,Y
552  NEXT I
555  INVERSE : PRINT T$;: NORMAL
560  PRINT "   CONTINUE ?";: CALL  - 756: TEXT : HOME
565  INPUT "ANY CHANGES IN SCALE FACTORS (Y/N) ? ";A$
570  IF A$ = "Y" THEN 455
575  RETURN
```

Fig. 3-20 (cont). Listing for DAMPED program.

We have the choice to compute the Fourier transform from either by entering discrete data points describing the waveform, or mathematically defining the function. Lines 210 to 235 describe, as an example, a rectangular pulse having a duty cycle of 87.5%, so that the amplitude is 1 for 7/8 of the period and zero for the remainder. After the input function is processed, the Fourier transform is tabulated in terms of its real and imaginary components, as well as its magnitude for each harmonic starting with the fundamental (i.e., harmonic "0"). In addition, these values are displayed in groups of 20. Pressing the RETURN key continues the listing until N/2 harmonics are displayed. At the end of the listing, hitting the RETURN key once more gives the magnitude spectrum plot as a function of its harmonic number.

Example 3-7

Compute the discrete Fourier transform for the function already described in Fig. 3-14, up to and including the 64th harmonic.

Fig. 3-15 shows the output results for the first 9 harmonics plus the fundamental while the spectrum plot is shown in Fig. 3-16. Fig. 3-17 shows the spectrum plot for a 25% duty cycle pulse (64 harmonics).

ANALYSIS OF DAMPED OSCILLATIONS

When an underdamped second-order system encounters a sudden change at its input, such as the voltage controlled oscillator of a phase-locked loop shifting from one frequency to another, the output tries to follow this change, but oscillates about a steady-state value for a time and eventually settles out. This process is illustrated in Fig. 3-18 for an impulse response, and in Fig. 3-19 for a step response. How fast this process is completed depends on the damping factor of the system. For smaller values of damping, it takes longer for the oscillations to settle down.

The DAMPED program listing in Fig. 3-20 determines the damping factor, the natural and damped frequencies. In addition, the program determines the normalized amplitude of the waveform at a given time. The required inputs are the positive amplitudes of two consecutive peaks, which will be one cycle apart, and the time period for one cycle. If desired, the resulting damped oscillation is plotted on the video display.

Example 3-8

From oscilloscope measurements of a damped oscillation resulting from a step response, it was measured that two consecutive peaks, like those of Fig. 3-19, had amplitudes of 8 and 3 divisions, respectively. In addition, the period for one complete cycle was measured as 0.46 ms. We want to know what will the amplitude be after 1.5 ms, when the peak input is 0.253 volt. Fig. 3-21 shows the results of the analysis.

```
AMPLITUDES ARE MEASURED FROM THE STEADY
STATE VALUE


POSITIVE AMPLITUDE OF 1ST CYCLE ? 8
POSITIVE AMPLITUDE OF 2ND CYCLE ? 3
CYCLE PERIOD IN SECONDS ? 0.46E-3

THEN:
DAMPING FACTOR = .156
NATURAL FREQUENCY = 2173.9 HZ
   (13659.1 RAD/SEC)
DAMPED NATURAL FREQ = 2147.9 HZ
   (13495.7 RAD/SEC)

DO YOU WISH TO KNOW THE RESPONSE AT A
SPECIFIC TIME  --  OR HAVE A GRAPH OF
THE WAVEFORM  ?

T)IME, G)RAPH OR Q)UIT? T
IS THE RESPONSE THE RESULT OF A:

S)TEP OR I)MPULSE INPUT ?I

TIME ? 1.5E-3
AMPLITUDE = 0.422664465

DO YOU WISH TO KNOW THE RESPONSE AT A
SPECIFIC TIME  --  OR HAVE A GRAPH OF
THE WAVEFORM  ?

T)IME, G)RAPH OR Q)UIT? G
IS THE RESPONSE THE RESULT OF A:

S)TEP OR I)MPULSE ? I

ENTER MIN,MAX TIME ?0,2E-3

MIN-MAX X VALUES:
0              2E-03
MIN-MAX Y VALUES:
-.490691       .799867

DESIRED MIN,MAX X-AXIS ?0,2E-3
DESIRED MIN,MAX Y-AXIS ?-.5,1
```

Fig. 3-21. Analysis of DAMPED oscillations.

4

Basic Statistics and the Analysis of Data

In any experimental endeavor, there usually comes a time when we must ask ourselves whether or not the data is a true representation of the experiment. Although this chapter is not meant to be a course in statistics, it is nevertheless important to include several useful statistical methods. Included are programs for computing the mean and standard deviation of a set of data, and regression analysis.

LEAST SQUARES REGRESSION CURVES

Almost every experimenter seeks to establish a relationship between two or more variables, such as the variation of impedance with frequency, or the variation of a transistor operating point as a function of ambient temperature and its current gain. As an aid in establishing a relationship (or lack of it), it is often desirable to express the relationship in the form of an equation.

Relationship Between Two Variables

Listed in Chart 4-1 are several common forms of mathematical relationships used to relate the behavior of two variables. To decide which equation should be used, it is helpful to graph the data in the form of a scatter diagram, as shown in Fig. 4-1, that can be compared against one of the general curves shown in Fig. 4-2.

Fig. 4-3 lists the REGRESSN program for determining the least squares regression of any one

Chart 4-1. Common Forms of Mathematical Relationships

Linear	$y = mx + b$
Inverse	$y = 1/(mx + b)$
Nth order (polynomial)	$y = a_0 + a_1x + a_2x^2 + \ldots$
Geometric	$y = ax^b$
Exponential	$y = ae^{bx}$

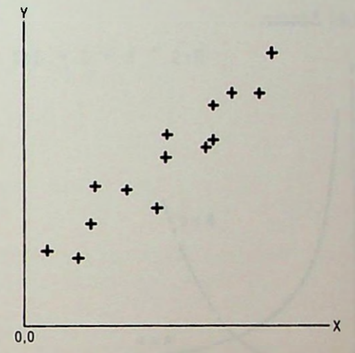
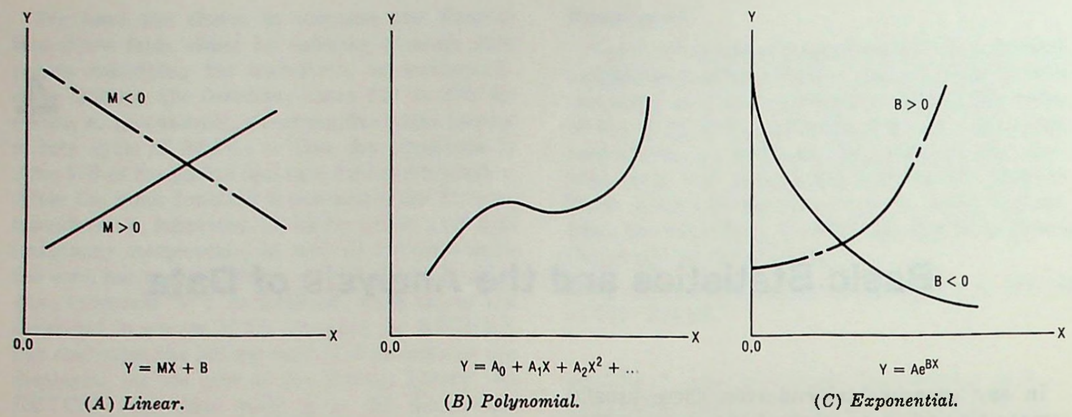


Fig. 4-1. Scatter diagram.

of the general forms listed in Chart 4-1. The REGRESSN program does the following:

1. Stores and retrieves data from disk.
2. Adds, deletes, or changes data.
3. Determines the minimum and maximum values of data points.
4. Plots a scatter diagram of the entered data.
5. Determines the number of data points entered.
6. Computes the coefficient of determination, correlation coefficient, and standard error of estimate.
7. Interpolates.

The REGRESSN program, like the plotting programs of Chapter 2, uses HiRes graphics for plotting the scatter diagram. The program is loaded at the decimal memory location 2048 and is approximately 5967 bytes long. It must be remembered that the HiRes graphics begins at decimal memory location 8192, so that there are only approximately 177 free bytes for adding lines to



(A) Linear. *(B) Polynomial.* *(C) Exponential.*

(D) Geometric. *(E) Inverse.*

Fig. 4-2. Generalized regression curves.

the program before REGRESSN overlaps the HiRes display area.

You may want to delete the REM lines from REGRESSN to enable you to add more lines to it. Before running the modified program, you may check it by typing:

```
PRINT PEEK(105) + PEEK(106)*256
```

and then hitting the RETURN key. The number you will see is the current LOMEM. If LOMEM is greater than 8192, then you must delete some of your program and try again. Remember, LOMEM in Applesoft is the end of the program area and is pushed up or down as you add or delete lines.

Example 4-1

Determine what kind of relationship exists between the following 10 X-Y data points, which are the parameters of voltage in millivolts, and temperature in degrees Fahrenheit for a given copper-constantan thermocouple: (−2.581, −100), (−1.101, −20), (−0.674, 0), (−0.404, 13), (−0.150, 25), (−0.043, 30), (0, 31), (1.086, 80), (1.542, 101), and (3.711, 190).

When the REGRESSN program is run, we are first asked if we want to enter data that has been previously stored on disk. If the above data has not been previously stored, the typed response is the letter N, in which case we are asked to enter each point, as shown in Fig. 4-4. After all data points have been entered, hit the RETURN key to terminate. All the data points are then listed so that we may then examine them for any possible errors or omissions. For example, suppose point No. 7 was supposed to be (0, 32) instead of (0, 31). We then type the letter Y in response to whether or not we want to change a data point.


```
  0 LOMEM: 16384: REM    BEFORE   SETTING ANY VARIABLES
  1 REM  **** CAUTION ****
100 REM ADDING TO THIS PROGRAM
101 REM MAY CAUSE PROBLEMS
102 REM   FROM HIRES PAGE1
103 REM  OVERLAYING PROGRAM
104 TEXT :D$ =  CHR$ (13) +  CHR$ (4)
105 HOME : VTAB 6: PRINT  TAB( 6): INVERSE : PRINT "LINEAR & CURVILINEAR
    REGRESSION": NORMAL
106 VTAB 20:Z9 = 1
107 PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
108 VTAB (13): PRINT  SPC( 11)"APPLESOFT VERSION"
109 FOR II = 1 TO 2500: NEXT
110 DIM X(100),Y(100),P(100,2),V1(100),A(21),U(11,12),T(12)
111 AI$ = "ANOTHER INTERPOLATION (Y/N) "
112 GOSUB 290: GOSUB 224: GOSUB 241
113 GOTO 257
114 HOME : PRINT "LINEAR REGRESSION"
115 GOSUB 315
116 FOR I = 1 TO N:X = X(I):Y = Y(I):J = J + X:K = K + Y:L = L + X ^ 2:M
    = M + Y ^ 2:R2 = R2 + X * Y: NEXT I
117 GOSUB 263
118 PRINT : PRINT "Y = ";A;" + (";B;" * X)"
119 GOSUB 265
120 INPUT "INTERPOLATION (Y/N) ";C$
121 IF C$ <  > "Y" THEN 126
122 INPUT "X= ";X
123 PRINT "    THEN Y = ";A + B * X
124 PRINT AI$;: INPUT C$
125 IF C$ = "Y" THEN 122
126 INPUT "SAVE INPUT DATA TO DISK (Y/N) ";ZZ$
127 IF ZZ$ = "Y" THEN  GOSUB 274
128 GOTO 262
129 GOSUB 315
130 FOR I = 1 TO N:X = X(I):Y = Y(I):Y =  LOG (Y):X =  LOG (X):J = J + X
    :K = K + Y:L = L + X ^ 2:M = M + Y ^ 2:R2 = R2 + X * Y: NEXT I
131 GOSUB 263
132 HOME : PRINT "GEOMETRIC REGRESSION (FOR X,Y>0)": PRINT
133 PRINT "      B": PRINT "Y = AX"
134 PRINT : PRINT "     A = "; EXP (A)
135 PRINT "     B = ";B
136 GOSUB 265
137 INPUT "INTERPOLATION (Y/N) ";C$
138 IF C$ <  > "Y" THEN 143
139 INPUT "X = ";X
140 PRINT "     THEN Y = "; EXP (A) * (X ^ B)
141 PRINT AI$;: INPUT C$: IF C$ <  > "Y" THEN 143
142 GOTO 139
143 INPUT "SAVE DATA TO DISK (Y/N) ";ZZ$
144 IF ZZ$ = "Y" THEN  GOSUB 274
145 GOTO 262
146 GOSUB 315
147 FOR I = 1 TO N:X = X(I):Y = Y(I):Y =  LOG (Y):J = J + X:K = K + Y:L =
    L + X ^ 2:M = M + Y ^ 2:R2 = R2 + X * Y: NEXT I: GOSUB 263
148 HOME : PRINT "EXPONENTIAL REGRESSION (FOR Y<>0)": PRINT
149 PRINT "         BX"
150 PRINT "Y = A(EXP)"
151 PRINT : PRINT "     A ="; EXP (A): PRINT "     B =";B
152 GOSUB 265
```

Continued on next page.

Fig. 4-3. Listing for REGRESSN program.

```
153  INPUT "INTERPOLATION (Y/N) ";C$
154  IF C$ <  > "Y" THEN 159
155  INPUT "X = ";X
156  PRINT "     THEN Y = "; EXP (A) *  EXP (B * X)
157  PRINT AI$;: INPUT C$: IF C$ <  > "Y" THEN 159
158  GOTO 155
159  INPUT "SAVE DATA TO DISK (Y/N) ";ZZ$
160  IF ZZ$ = "Y" THEN  GOSUB 274
161  GOTO 262
162  HOME : PRINT "POLYNOMIAL REGRESSION"
163  PRINT  SPC( 22)"N"
164  PRINT "Y = A  + A X +.....A X"
165  PRINT "     0    1         N"
166  PRINT : INPUT "ORDER OF POLYNOMIAL ";D
167  HOME : VTAB (9): PRINT  TAB( 15): FLASH : PRINT "CALCULATING...": NORMAL
168  A(1) = N:J = 0:K = 0:Q = 0:S = 0:Z = 0:I = 0:CC = 0
169  FOR I = 1 TO N
170  X = X(I):Y = Y(I)
171  FOR J = 2 TO 2 * D + 1:A(J) = A(J) + X ^ (J - 1): NEXT J
172  FOR K = 1 TO D + 1:U(K,D + 2) = T(K) + Y * X ^ (K - 1):T(K) = T(K) +
     Y * X ^ (K - 1): NEXT K
173  T(D + 2) = T(D + 2) + Y ^ 2: NEXT I
174  FOR J = 1 TO D + 1: FOR K = 1 TO D + 1:U(J,K) = A(J + K - 1): NEXT K
     : NEXT J
175  FOR J = 1 TO D + 1: FOR K = J TO D + 1
176  IF U(K,J) <  > 0 THEN 179
177  NEXT K
178  INVERSE : PRINT "NO UNIQUE SOLUTION CAN BE FOUND": NORMAL : GOTO 206
179  FOR I = 1 TO D + 2:S = U(J,I):U(J,I) = U(K,I):U(K,I) = S: NEXT I
180  Z = 1 / U(J,J)
181  FOR I = 1 TO D + 2:U(J,I) = Z * U(J,I): NEXT I
182  FOR K = 1 TO D + 1
183  IF K = J THEN 186
184  Z =  - U(K,J)
185  FOR I = 1 TO D + 2:U(K,I) = U(K,I) + Z * U(J,I): NEXT I
186  NEXT K: NEXT J
187  HOME : PRINT "POLYNOMIAL COEFFICIENTS:": PRINT "A(0) = ";U(1,D + 2)
188  FOR J = 1 TO D: PRINT "A(";J;") = ";U(J + 1,D + 2): NEXT J
189  P = 0: FOR J = 2 TO D + 1:P = P + U(J,D + 2) * (T(J) - A(J) * T(1) /
     N): NEXT J
190  Q = T(D + 2) - T(1) ^ 2 / N:Z = Q - P:I = N - D - 1:L = P / Q
191  PRINT : INPUT "HIT RETURN TO CONTINUE ";C$
192  PRINT  SPC( 21)"2"
193  PRINT "DETERMINATION COEFF R = ";L
194  PRINT "CORRELATION COEFFICIENT R = "; SQR (L)
195  PRINT "STANDARD ESTIMATE ERROR = "; SQR (Z / I)
196  INPUT "INTERPOLATION (Y/N) ";C$
197  IF C$ <  > "Y" THEN 204
198  INPUT "X = ";X
199  P = U(1,D + 2)
200  FOR J = 1 TO D:P = P + U(J + 1,D + 2) * X ^ J: NEXT J
201  PRINT "     THEN Y = ";P
202  PRINT AI$;: INPUT C$
203  IF C$ = "Y" THEN 198
204  INPUT "SAVE INPUT DATA TO DISK (Y/N) ";ZZ$
205  IF ZZ$ = "Y" THEN  GOSUB 274
206  :
207  GOTO 262
```

Continued on next page.

Fig. 4-3 (cont). Listing for REGRESSN program.


```
208  HOME : PRINT "INVERSE REGRESSION"
209  GOSUB 315
210  FOR I = 1 TO N:X = 1 / X(I):Y = Y(I)
211  J = J + X:K = K + Y:L = L + X ^ 2:M = M + Y ^ 2:R2 = R2 + X * Y: NEXT
     I
212  GOSUB 263
213  PRINT : PRINT "Y = ";A;" + (";B;"/X)"
214  GOSUB 265
215  INPUT "INTERPOLATION (Y/N) ";C$
216  IF C$ <  > "Y" THEN 221
217  INPUT "X = ";X
218  PRINT "     THEN Y = ";A + B / X
219  PRINT AI$;: INPUT C$
220  IF C$ = "Y" THEN 217
221  INPUT "SAVE DATA TO DISK (Y/N) ";ZZ$
222  IF ZZ$ = "Y" THEN  GOSUB 274
223  GOTO 262
224  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
225  FOR I = 1 TO N
226  IF (Y1 - Y(I)) <  = 0 THEN 228
227  Y1 = Y(I): GOTO 230
228  IF (Y2 - Y(I)) >  = 0 THEN 230
229  Y2 = Y(I)
230  IF (X1 - X(I)) <  = 0 THEN 233
231  X1 = X(I)
232  GOTO 235
233  IF (X2 - X(I)) >  = 0 THEN 235
234  X2 = X(I)
235  NEXT I
236  HOME : PRINT "YOUR DATA HAS THE FOLLOWING LIMITS:"
237  PRINT : PRINT : PRINT : PRINT "VARIABLE","MINIMUM","MAXIMUM"
238  PRINT "X",X1,X2: PRINT "Y",Y1,Y2
239  XD = X2 - X1:YD = Y2 - Y1
240  RETURN
241  PRINT : INPUT "HIT RETURN FOR A GRAPH OF INPUT DATA";FF$
242  HGR : HCOLOR= 7: VTAB (23)
243  HPLOT 8,1 TO 8,150
244  FOR Y = 150 TO 0 STEP  - (150 / 8): HPLOT 6,Y TO 7,Y: NEXT Y
245  FOR Y = 150 TO 0 STEP  - (150 / 4): HPLOT 3,Y TO 7,Y: NEXT Y
246  HPLOT 13,154 TO 279,154
247  FOR X = 13 TO 279 STEP 266 / 8: HPLOT X,155 TO X,157: NEXT X
248  FOR X = 13 TO 279 STEP 266 / 4: HPLOT X,157 TO X,159: NEXT X
249  VTAB 24: PRINT "Y-AXIS ";Y1;":";Y2; SPC( 5)"X-AXIS ";X1;":";X2: PRINT
250  FOR I = 1 TO N
251  X = 279 - ((X2 - X(I)) * 266 / XD)
252  Y = 150 - ((Y(I) - Y1) * 150 / YD)
253  IF X > 279 OR X < 13 OR Y < 0 THEN 5290
254  YT = Y
255  HPLOT X,Y: NEXT I
256  PRINT "HIT RETURN TO CONTINUE ";: CALL  - 756: TEXT : HOME : RETURN
257  HOME : PRINT "REGRESSION TYPES AVAILABLE:"
258  PRINT : PRINT "1)  LINEAR REGRESSION": PRINT "2)  INVERSE REGRESSION
     ": PRINT "3)  NTH ORDER (POLYNOMIAL) REGRESSION": PRINT "4)  GEOMETR
     IC REGRESSION": PRINT "5)  EXPONENTIAL REGRESSION": PRINT : PRINT
259  INPUT "YOUR CHOICE? ";ZX
260  IF ZX < 1 OR ZX > 5 THEN 257
261  ON ZX GOTO 114,208,162,129,146
```

Continued on next page.

Fig. 4-3 (cont). Listing for REGRESSN program.

```
262  PRINT : END
263  B = (N * R2 - K * J) / (N * L - J ^ 2)
264  A = (K - B * J) / N: RETURN
265  J = B * (R2 - J * K / N)
266  M = M - K ^ 2 / N
267  K = M - J: IF K < 0 THEN K = 0
268  R2 = J / M: PRINT : PRINT "NUMBER OF DATA POINTS = ";N
269  PRINT  SPC( 21)"2"
270  PRINT "DETERMINATION COEFF R = ";R2
271  PRINT "CORRELATION COEFFICIENT R = "; SQR (R2)
272  PRINT "STANDARD ESTIMATE ERROR = "; SQR (K / (N - 2))
273  RETURN
274  REM  SAVE DATA
275  PRINT : INPUT "FILENAME ? ";NM$
276  PRINT D$;"OPEN ";NM$
277  PRINT D$;"WRITE";NM$
278  PRINT N
279  FOR I = 1 TO N: PRINT X(I): PRINT Y(I): NEXT I
280  PRINT D$;"CLOSE";NM$
281  RETURN
282  REM  LOAD DATA
283  PRINT : INPUT "FILENAME ? ";NM$
284  IF NM$ = "" THEN 283
285  IF  ASC (NM$) < 65 THEN 283
286  PRINT D$;"OPEN ";NM$: PRINT D$;"READ ";NM$: INPUT N
287  FOR I = 1 TO N: INPUT X(I): INPUT Y(I): NEXT I
288  PRINT D$;"CLOSE";NM$
289  RETURN
290  HOME : INPUT "INPUT DATA POINTS FROM DISK (Y/N) ? ";ZZ$
291  IF ZZ$ <  > "Y" THEN 293
292  GOSUB 282: GOSUB 303: RETURN
293  HOME : PRINT "ENTER EACH DATA POINT (RETURN TO QUIT)": PRINT
294  PRINT  SPC( 8)"X-POINT"; SPC( 12)"Y-POINT": PRINT :N = 1
295  INPUT "X> ";XX$
296  IF XX$ = "" THEN 302
297  CALL  - 998: CALL  - 868
298  PRINT  TAB( 16 -  LEN (XX$)) VAL (XX$): CALL  - 998
299  HTAB (20): INPUT "Y> ";YY$
300  CALL  - 998: HTAB (20): CALL  - 868: HTAB (20): PRINT  SPC( 15 -  LEN
     (YY$)); VAL (YY$)
301  X(N) =  VAL (XX$):Y(N) =  VAL (YY$):N = N + 1: GOTO 295
302  N = N - 1
303  HOME : PRINT "POINT","X","Y"
304  FOR VV = 1 TO N: PRINT VV,X(VV),Y(VV): NEXT
305  PRINT : INPUT "CHANGE ANY DATA POINTS (Y/N) ";C$
306  IF  LEFT$ (C$,1) <  > "Y" THEN 310
307  GOSUB 316
308  INPUT "ANOTHER CHANGE (Y/N) ";C$.
309  IF C$ = "Y" THEN 307
310  INPUT "DELETE A DATA POINT (Y/N) ";DD$
311  IF DD$ = "Y" THEN 320
312  INPUT "DO YOU WANT TO ADD DATA POINTS (Y/N) ";C$
313  IF C$ <  > "Y" THEN  RETURN
314  N = N + 1: GOTO 295
315  I = 0:J = 0:K = 0:L = 0:M = 0:Q = 0:S = 0:Z = 0:CC = 0:R2 = 0: RETURN
316  HOME : PRINT "POINT","X","Y"
317  FOR VV = 1 TO N: PRINT VV,X(VV),Y(VV): NEXT VV
318  PRINT : INPUT "CHANGE POINT # ";CC
```

Continued on next page.

Fig. 4-3 (cont). Listing for REGRESSN program.


```
319  INPUT "NEW X,Y ";X(CC),Y(CC): RETURN
320  HOME : PRINT "POINT","X","Y"
321  FOR VV = 1 TO N: PRINT VV,X(VV),Y(VV): NEXT VV
322  PRINT : INPUT "DELETE POINT # ";KK
323  FOR L = KK TO N:X(L) = X(L + 1):Y(L) = Y(L + 1): NEXT L:N = N - 1: GOTO
     303
```

Fig. 4-3 (cont). Listing for REGRESSN program.

We are then asked which point is to be changed, and then add the new coordinates for that point.

After all corrections or additions have been made, the display then informs us of the minimum and maximum limits of our data and then displays a scatter diagram (Fig. 4-5), which can then be compared against one of the generalized curves shown in Fig. 4-2. For this example, the linear, or straight line, regression appears to be the best choice. Fig. 4-6 shows the results for the linear regression line of our data, which includes the interpolation for $x = -0.171$. If no other interpolations are required, we then decide whether or not this data is to be saved on disk.

```
INPUT DATA POINTS FROM DISK (Y/N) ? N
ANTER EACH DATA POINT (RETURN TO QUIT)

          X-POINT              Y-POINT
           -2.581                 -100
           -1.101                  -20
            -.674                    0
            -.404                   13
             -.15                   25
            -.043                   30
                0                   31
            1.086                   80
            1.542                  101
            3.711                  190
X>

POINT       X                       Y
1           -2.581                  -100
2           -1.101                  -20
3           -.674                   0
4           -.404                   13
5           -.15                    25
6           -.043                   30
7           0                       31
8           1.086                   80
9           1.542                   101
10          3.711                   190

CHANGE ANY DATA POINTS (Y/N) Y
CHANGE POINT # 7
NEW X,Y 0,32
ANOTHER CHANGE (Y/N) N
DELETE A DATA POINT (Y/N) N
DO YOU WANT TO ADD DATA POINTS (Y/N) N

YOUR DATA HAS THE FOLLOWING LIMITS:

VARIABLE          MINIMUM           MAXIMUM
X                 -2.581            3.711
Y                 -100              190

HIT RETURN FOR A GRAPH OF INPUT DATA
```

Fig. 4-4. Input data for Example 4-1.

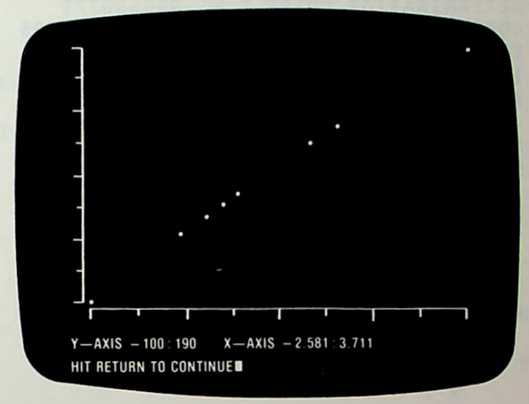


Fig. 4-5. Video scatter diagram of Example 4-1.

Because of the method in which the remaining four types of regression curves are done, the following restrictions are required:

Inverse—any x cannot equal 0.
Polynomial—order of polynomial cannot be greater than the number of data points less 2.
Geometric—all x's and y's must be greater than 0.
Exponential—any y cannot equal 0.

Table 4-1. Data for Example 4-2

Thermal Resistance (degrees/watt)	Volume (in^3)
12.0	1.0
10.0	1.3
8.0	2.0
4.0	6.0
1.3	30.0
0.8	70.0
0.4	200
0.3	300
0.13	900

```
LINEAR REGRESSION

Y = 28.7844229 + (45.5669342 * X)

NUMBER OF DATA POINTS = 10
                      2
DETERMINATION COEFF R = .995586779
CORRELATION COEFFICIENT R = .99779095
STANDARD ESTIMATE ERROR = 5.432569
INTERPOLATION (Y/N) Y
X= -.171
    THEN Y = 20.9924772
ANOTHER INTERPOLATION (Y/N) ?N
SAVE INPUT DATA TO DISK (Y/N) Y

FILENAME ? SAMPLE 1
```

Fig. 4-6. Linear regression output results.

```
POLYNOMIAL REGRESSION
                        N
Y = A  + A X +...   A X
     0      1        N

ORDER OF POLYNOMIAL 3

POLYNOMIAL COEFFICIENTS:
A(0) = 32.0932221
A(1) = 46.1803591
A(2) = -1.52832688
A(3) = .149673052

HIT ENTER TO CONTINUE
                      2
DETERMINATION COEFF R = .999966237
CORRELATION COEFFICIENT R = .999983019
STANDARD ESTIMATE ERROR = .148678266
INTERPOLATION (Y/N) Y
X = -.171
     THEN Y = 24.1509425
ANOTHER INTERPOLATION (Y/N) ?N
```

Fig. 4-7. Third-order polynomial regression output results.

Although we see that it is possible to express our data in terms of a straight line, there may be another form that is better suited. For example, it might be possible that our original data may be expressed in the form of an n-th order polynomial curve. Fig. 4-7 shows the results for the polynomial regression, using a third-order approximation. We see that the standard error is less, indicating that this might be the better form. When regression curves are compared, statistical tables should be consulted to determine whether or not the correlation coefficient is significant for the number of data points used.

```
GEOMETRIC REGRESSION (FOR X,Y>0)
       B
Y = AX

     A = 46.1540021
     B = -1.52145322

NUMBER OF DATA POINTS = 9
                      2
DETERMINATION COEFF R = .999151341
CORRELATION COEFFICIENT R = .99957558
STANDARD ESTIMATE ERROR = .0793129252
INTERPOLATION (Y/N) Y
X = 4.0
     THEN Y = 5.60031765
ANOTHER INTERPOLATION (Y/N) ?N
```

Fig. 4-8. Geometric regression results.

Example 4-2

Table 4-1 summarizes the data points from an experiment that measured the thermal resistance of a heat sink as a function of its volume. From past experience we might know that this should look like a straight line when plotted using log-log coordinates, thus suggesting a geometric regression.

Fig. 4-8 shows the output for the data of Table 4-1 using a geometric regression.

Relationship Between More Than Two Variables

The REGRESSN program dealt only with the possible relationships that may exist between a dependent variable and a single independent variable. However, there are situations where the measured quantity is dependent upon two or more quantities, which may or may not be interrelated.

The multiple linear regression program (MLREGRES) listed in Fig. 4-9 determines the coefficients for the generalized form:

$$y = a_0 + a_1x_1 + a_2x_2 \ldots + a_Nx_N$$

In addition, the program determines the coefficients of determination and correlation, as well as the standard error of estimate.

Example 4-3

For the following data, determine the multiple regression coefficients, if a solution can be found.

dependent variable y	64	71	53	67	55	58
independent variable x_1	57	59	49	62	51	50
independent variable x_2	8	10	6	11	8	7


```
100  REM  MULTIPLE LINEAR REGRESSION (MLREGRES)
101  HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "MULTIPLE LINEAR RE
     GRESSION": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR I = 1 TO 2500: NEXT I: HOME
105  DIM X(30),S(30),T(30),A(30,31)
106  PRINT : INPUT "NUMBER OF DATA POINTS? ";N
107  PRINT : INPUT "NUMBER OF INDEPENDENT VARIABLES? ";V
108  HOME
109  PRINT : PRINT "ENTER DEPENDENT & INDEPENDENT VARIABLES": PRINT "FOR
     EACH DATA POINT"
110  PRINT :X(1) = 1
111  FOR I = 1 TO N: PRINT "POINT ";I
112  INPUT "  DEPENDENT VARIABLE Y? ";X(V + 2)
113  FOR J = 1 TO V
114  PRINT "  INDEPENDENT VARIABLE A(";J;") ";
115  INPUT X(J + 1)
116  NEXT J
117  FOR K = 1 TO V + 1
118  FOR L = 1 TO V + 2
119  A(K,L) = A(K,L) + X(K) * X(L)
120  S(K) = A(K,V + 2)
121  NEXT L
122  NEXT K
123  S(V + 2) = S(V + 2) + X(V + 2) ^ 2
124  NEXT I
125  FOR I = 2 TO V + 1
126  T(I) = A(1,I)
127  NEXT I
128  FOR I = 1 TO V + 1
129  FOR J = I TO V + 1
130  IF A(J,I) <  > 0 THEN 134
131  NEXT J
132  PRINT : INVERSE : PRINT "NO UNIQUE SOLUTION": NORMAL
133  GOTO 180
134  FOR K = 1 TO V + 2
135  B = A(I,K)
136  A(I,K) = A(J,K)
137  A(J,K) = B
138  NEXT K
139  Z = 1 / A(I,I)
140  FOR K = 1 TO V + 2
141  A(I,K) = Z * A(I,K)
142  NEXT K
143  FOR J = 1 TO V + 1
144  IF J = I THEN 149
145  Z =  - A(J,I)
146  FOR K = 1 TO V + 2
147  A(J,K) = A(J,K) + Z * A(I,K)
148  NEXT K
149  NEXT J
150  NEXT I
151  HOME : PRINT : PRINT "RESULTS OF MULTIPLE LINEAR REGRESSION:": PRINT
152  PRINT "A(0) = ";A(1,V + 2)
153  FOR I = 2 TO V + 1
```

Continued on next page.

Fig. 4-9. Listing for MLREGRES program.

```
154  PRINT "A(";I - 1;") = ";A(I,V + 2)
155  NEXT I
156  P = 0
157  FOR I = 2 TO V + 1
158  P = P + A(I,V + 2) * (S(I) - T(I) * S(1) / N)
159  NEXT I
160  R = S(V + 2) - S(1) ^ 2 / N
161  Z = R - P
162  L = N - V - 1
163  I = P / V
164  I = P / R
165  PRINT : PRINT  TAB( 31)"2"
166  PRINT "COEFFICIENT OF DETERMINATION R = "; INT (I * 1E6 + .5) / 1E6
167  PRINT : PRINT "CORRELATION COEFFICIENT R = "; INT ( SQR (I) * 1E6 +
     .5) / 1E6
168  PRINT : PRINT "STANDARD ERROR OF ESTIMATE = "; INT ( SQR ( ABS (Z /
     L)) * 1E6 + .5) / 1E6
169  PRINT : INPUT "INTERPOLATION (Y/N)? ";C$
170  IF  LEFT$ (C$,1) = "Y" THEN 172
171  GOTO 180
172  P = A(1,V + 2)
173  FOR J = 1 TO V
174  PRINT : PRINT "INDEPENDENT VARIABLE A(";J;") ";
175  INPUT X
176  P = P + A(J + 1,V + 2) * X: NEXT J
177  PRINT : PRINT "  THEN Y = ";P
178  PRINT : INPUT "ANOTHER INTERPOLATION ? ";C$
179  IF C$ = "Y" THEN 172
180  PRINT : END
```

Fig. 4-9 (cont). Listing for MLREGRES program.

As shown in the output results of Fig. 4-10, the dependent variable is entered first for each point, followed by the two independent variables x_1 and x_2. The regression coefficients then yield the resulting equation:

$$y = -5.03124863 + 1.21874996x_1 - 0.0312499053x_2$$

In addition, the interpolation is shown for the values $x_1 = 52$ and $x_2 = 10$.

SAMPLE STATISTICS

The DSTATS program listed in Fig. 4-11 determines the arithmetic mean, variance, standard deviation, skewness, kurtosis, Z-scores, maximum and minimum values, and the range for a sample of ungrouped data. The variance and standard deviation is a measure of dispersion of the data about the mean value. The coefficients of skewness and kurtosis are measures of asymmetry and peakness respectively of a distribution of data. The Z-scores are measures of the deviation from the mean in terms of a standard deviation, so that other distributions of data may be compared.

When run, the DSTATS program displays the value, its difference from the mean, and Z-score in groups of 20. Hitting the RETURN key continues the rest of the list. After the entire list of values is displayed, the remaining statistics of the sample are given.

Example 4-4

Determine the sample statistics for the following 15 measured values for 100-ohm resistors: 99, 103, 99, 98, 100, 101, 105, 97, 99, 100, 102, 101, 103, 98, and 101.

As shown in Fig. 4-12, the mean value for these 15 resistors over the range of 97 to 105 is 100.4 ohms, while the variance and standard deviation are 4.828562 and 2.197399 ohms, respectively.

Fig. 4-10. Multiple linear regression results.

```
NUMBER OF DATA POINTS? 6

NUMBER OF INDEPENDENT VARIABLES? 2

ENTER DEPENDENT & INDEPENDENT VARIABLES
FOR EACH DATA POINT

POINT 1
  DEPENDENT VARIABLE Y? 64
  INDEPENDENT VARIABLE A(1) ?57
  INDEPENDENT VARIABLE A(2) ?8
POINT 2
  DEPENDENT VARIABLE Y? 71
  INDEPENDENT VARIABLE A(1) ?59
  INDEPENDENT VARIABLE A(2) ?10
POINT 3
  DEPENDENT VARIABLE Y? 53
  INDEPENDENT VARIABLE A(1) ?49
  INDEPENDENT VARIABLE A(2) ?6
POINT 4
  DEPENDENT VARIABLE Y? 67
  INDEPENDENT VARIABLE A(1) ?62
  INDEPENDENT VARIABLE A(2) 11
POINT 5
  DEPENDENT VARIABLE Y? 55
  INDEPENDENT VARIABLE A(1) ?51
  INDEPENDENT VARIABLE A(2) ?8
POINT 6
  DEPENDENT VARIABLE Y? 58
  INDEPENDENT VARIABLE A(1) ?50
  INDEPENDENT VARIABLE A(2) ?7

RESULTS OF MULTIPLE LINEAR REGRESSION:

A(0) = -5.03124863
A(1) = 1.21874996
A(2) = -.0312499053

                                  2
COEFFICIENT OF DETERMINATION R = .838158
CORRELATION COEFFICIENT R = .91551
STANDARD ERROR OF ESTIMATE = 3.696845

INTERPOLATION (Y/N)? Y

INDEPENDENT VARIABLE A(1) ?52
INDEPENDENT VARIABLE A(2) ?10
  THEN Y = 58.0312503

ANOTHER INTERPOLATION ? N
```

```
100 REM  DESCRIPTIVE STATISTICS (DSTATS)
101 CLEAR : HOME : VTAB (7): PRINT  TAB( 5): INVERSE : PRINT "DESCRIPTIV
    E STATISTICS PROGRAM": NORMAL
102 PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
    HOWARD M. BERLIN, LTD."
103 PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
    VERSION"
104 FOR I = 1 TO 2500: NEXT : HOME
105 DIM N(100): FOR I = 1 TO 39:LI$ = LI$ + "=": NEXT
106 PRINT : INPUT "NUMBER OF OBSERVATIONS? ";N: PRINT
107 M = 0:B = 0:C = 0:T = 0:U = 0
108 FOR I = 1 TO N: PRINT "#";I;
109 INPUT " ";X:N(I) = X
110 C = C + X:B = B + X ^ 2: NEXT I
111 M = C / N:V = (B - N * M ^ 2) / (N - 1):T = V * (N - 1) / N:U =  SQR
    (T)
112 D =  SQR (V)
113 M =  INT (M * 1E6 + .5) / 1E6:V =  INT (V * 1E6 + .5) / 1E6:D =  INT
    (D * 1E6 + .5) / 1E6
114 PRINT : PRINT "HIT RETURN FOR LISTINGS OF": PRINT "VALUES AND Z-SCOR
    ES ";
115 CALL  - 756
116 P = 0:Q = 0:R = 0:I = 1:S = 1
117 HOME
118 HOME : PRINT : PRINT "#  VALUE    DIFFERENCE       Z-SCORE": PRINT L
    INE$
119 P = N(I) - M:Z = P / D
120 Q = Q + (P ^ 3):R = R + (P ^ 4)
121 P =  INT (P * 1E6 + .6) / 1E6:Z =  INT (Z * 1E6 + .5) / 1E6
122 PRINT I; TAB( 5)N(I); TAB( 14)P; TAB( 30)Z
123 IF I = N THEN 129
124 IF I > (20 * S) - 1 THEN 126
125 I = I + 1: GOTO 119
126 POKE  - 16368,0:S = S + 1: PRINT "  -- HIT ANY KEY TO CONTINUE -- ";
127 CALL  - 756
128 I = I + 1: GOTO 118
129 PRINT "**  END OF LIST **"
130 PRINT "HIT RETURN FOR SAMPLE STATISTICS ";
131 CALL  - 756
132 N1 = N(1):N2 = N(1)
133 FOR I = 2 TO N
134 IF (N1 - N(I)) <  = 0 THEN 136
135 N1 = N(I): GOTO 138
136 IF (N2 - N(I)) >  = 0 THEN 138
137 N2 = N(I)
138 NEXT I
139 HOME : PRINT : PRINT "MINIMUM VALUE = ";N1: PRINT : PRINT "MAXIMUM V
    ALUE = ";N2: PRINT : PRINT "RANGE = "; ABS (N2 - N1)
140 PRINT : PRINT "MEAN = ";M: PRINT : PRINT "VARIANCE = ";V: PRINT : PRINT
    "STD. DEVIATION = ";D
141 PRINT : PRINT "SKEWNESS = "; INT ((Q / (N * (U ^ 3)) * 1E6 + .5)) /
    1E6
142 PRINT : PRINT "KURTOSIS = "; INT ((R / (N * (U ^ 4)) * 1E6 + .5)) /
    1E6
143 PRINT : END
```

Fig. 4-11. Listing for DSTATS program.


```
NUMBER OF OBSERVATIONS? 15
#1 99
#2 103
#3 99
#4 98
#5 100
#6 101
#7 105
#8 97
#9 99
#10 100
#11 102
#12 101
#13 103
#14 98
#15 101

HIT RETURN FOR LISTINGS OF
VALUES AND Z SCORES


#  VALUE    DIFFERENCE       Z-SCORE
1    99       -1.4           -.637117
2    103      2.6            1.182217
3    99       -1.4           -.637117
4    98       2.4            -1.0922
5    100      -.4            -.182033
6    101      .6             .27705
7    105      4.6            2.093384
8    97       -3.4           -1.547284
9    99       -1.4           -.637117
10   400      -.4            -.182033
11   102      1.6            .728132
12   101      .6             .27705
13   103      2.6            1.182217
14   98       -2.4           -1.0922
15   101      .6             .27705

END OF LIST

MINIMUM VALUE = 97
MAXIMUM VALUE = 105
RANGE = 8
MEAN = 100.4
VARIANCE = 4.828562
STD. DEVIATION = 2.197399
SKEWNESS = .431235
KURTOSIS = 2.4851
```

Fig. 4-12. Results of sample statistics for 15 resistor values.

5

Networks and Transforms

This chapter presents programs for the design of a variety of attenuators and lossless pads, which are primarily used for impedance matching. In addition, programs are included which permit the analysis of 2-port networks, the inverse Laplace transform, real and imaginary roots of polynomials, Pi-Tee (Delta-Wye) transformations, and the solution of mesh and node equations.

ROOTS OF POLYNOMIALS

Very often the analysis of a network requires that the resultant polynomial be factored into distinct roots, which may be either real or complex. Such may be the case of solving a high order differential equation by Laplace transforms, or determining the pole locations of Butterworth and Chebyshev polynomials.

The PLYROOTS program shown in Fig. 5-1 determines the real or complex roots, if any exist, for any polynomial of the form:

$$P(x) = a_0 + a_1x + a_2x^2 \ldots + a_Nx^N \qquad \text{(Eq. 5-1)}$$

Naturally, the number of roots will be equal to the degree of the polynomial. It should be noted that third- and higher-order polynomials frequently have roots that are conjugate pairs in which the real parts are equal but the imaginary parts are of opposite sign. The coefficients of the polynomial are entered one at a time, starting with the constant term on up to the nth-order term.

Example 5-1

Determine the roots, if any exist, for the second-order polynomial:

$$P(x) = x^2 + x - 6$$

As shown in Fig. 5-2 the coefficients of the polynomial are entered one at a time, starting with the constant term on up to the second-order term. For this example, the roots were found to be equal to 2 and −3, both of which are real. This solution agrees with the solution obtained using the quadratic formula method for second-order equations.

Example 5-2

Determine the pole locations for the following fourth-order Butterworth polynomial:

$$P(s) = 1 + 2.613s + 3.414s^2 + 2.613s^3 + s^4$$

Shown in Fig. 5-3 are the output results. We see that all four roots are complex, and exist as two complex conjugate pairs. From these roots, we then know that the pole locations lie 22.5 degrees apart on the perimeter of a semicircle of unit length in the left half of the complex plane, as shown in Fig. 5-4.

THE INVERSE LAPLACE TRANSFORM

The inverse Laplace transform:

$$f(t) = \frac{1}{j2\pi}\int_{\sigma_1 - j\infty}^{\sigma_1 + j\infty} F(s)e^{st}\,ds \qquad \text{(Eq. 5-2)}$$

is frequently used for the solution of linear differential equations by transforming the problem from the frequency domain to the time varying response.

The inverse Laplace transform program (ILAPLACE) shown in Fig. 5-5 assumes that the frequency domain, or transfer function of the complex variable s ($s = j\omega$), is expressed by the following generalized form:

```
100  REM  REAL AND COMPLEX ROOTS OF A REAL POLYNOMIAL (PLYROOTS)
101  HOME : VTAB (7): PRINT  TAB( 10): INVERSE : PRINT "REAL & COMPLEX RO
     OTS": NORMAL
102  PRINT : PRINT  TAB( 10): INVERSE : PRINT "OF A REAL POLYNOMIAL": NORMAL
103  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN ,LTD."
104  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
105  FOR II = 1 TO 2500: NEXT II: HOME : DIM A(40),C(40),RR(20),RI(20)
106  PRINT : INPUT "ORDER OF POLYNOMIAL? ";O
107  PRINT : PRINT "ENTER COEFFICIENTS STARTING WITH": PRINT "THE CONSTAN
     T TERM": PRINT
108  FOR I = 1 TO O + 1
109  PRINT "COEFFICIENT A(";I - 1;") ";
110  INPUT A(I)
111  NEXT I
112  IT = O:N = O
113  IF A(N + 1) <  > O THEN 115
114  GOTO 117
115  IF N > O THEN 117
116  GOTO 206
117  NX = O
118  NY = N + 1
119  N2 = 1
120  KJ = N + 1
121  FOR L = 1 TO KJ
122  MT = KJ - L + 1
123  C(MT) = A(L)
124  NEXT L
125  REM   SET INITIAL VALUES
126  XO = .00500101:YO = .01000101
127  IN = O
128  X = XO
129  XO =  - 10 * YO:YO =  - 10 * X
130  REM   SET X AND Y TO CURRENT VALUE
131  X = XO:Y = YO
132  IN = IN + 1: GOTO 137
133  IT = 1
134  XP = X
135  YP = Y
136  REM   EVALUATE POLYNOMIAL & DERIVATIVES
137  IC = O
138  UX = O:UY = O:V = O:YT = O:XT = 1.00
139  U = C(N + 1)
140  IF U <  > O THEN 142
141  GOTO 186
142  FOR I = 1 TO N
143  L = N - I + 1
144  T = C(L)
145  X2 = X * XT - Y * YT:Y2 = X * YT + Y * XT
146  U = U + T * X2
147  V = V + T * Y2
148  F1 = I
149  UX = UX + F1 * XT * T
150  UY = UY - F1 * YT * T
151  XT = X2:YT = Y2: NEXT I
152  SQ = (UX ^ 2) + (UY ^ 2)
153  IF SQ <  > O THEN 155
```

Continued on next page.

Fig. 5-1. Listing of PLYROOTS program.


```
154  GOTO 177
155  DX = (V * UY - U * UX) / SQ
156  X = X + DX
157  DY =  - (U * UY + V * UX) / SQ
158  Y = Y + DY
159  IF  ABS (DY) +  ABS (DX) - 1E - 5 < O THEN 167
160  REM   STEP ITERATION COUNTER
161  IC = IC + 1
162  HOME : VTAB (9): PRINT  TAB( 15): FLASH : PRINT "COMPUTING": NORMAL
163  IF IC - 500 < O THEN 138
164  IF IT <  > O THEN 167
165  IF IN - 5 < O THEN 128
166  GOTO 206
167  FOR L = 1 TO NY
168  MT = KJ - L + 1
169  T = A(MT)
170  A(MT) = C(L)
171  C(L) = T: NEXT L
172  IP = N
173  N = NX
174  NX = IP
175  IF IT <  > O THEN 180
176  GOTO 133
177  IF IT <  > O THEN 179
178  GOTO 128
179  X = XP:Y = YP
180  IT = O
181  IF  ABS (Y) - (1E - 4 *  ABS (X)) < O THEN 189
182  AL = X + X
183  SQ = (X ^ 2) + Y ^ 2
184  N = N - 2
185  GOTO 193
186  X = O
187  NX = NX - 1
188  NY = NY - 1
189  Y = O
190  SQ = O
191  AL = X
192  N = N - 1
193  C(2) = C(2) + AL * C(1)
194  FOR L = 2 TO N
195  C(L + 1) = C(L + 1) + AL * C(L) - SQ * C(L - 1)
196  NEXT L
197  RI(N2) = Y:RR(N2) = X
198  N2 = N2 + 1
199  IF SQ <  > O THEN 201
200  GOTO 204
201  Y =  - Y
202  SQ = O
203  GOTO 197
204  IF N > O THEN 126
205  GOTO 209
206  INVERSE : PRINT : PRINT "UNABLE TO COMPUTE AFTER 500 ITERATIONS"
207  NORMAL
208  END
209  REM   OUTPUT RESULTS
210  HOME : PRINT : PRINT "FOR THE POLYNOMIAL EQUATION:"
211  PRINT : PRINT "O = ";A(1);" + ";A(2);"X ";
```

Continued on next page.

Fig. 5-1 (cont). Listing of PLYROOTS program.

```
212  FOR K = 3 TO 0 + 1
213  PRINT "    + ";A(K);"X^";K - 1
214  NEXT K
215  PRINT : PRINT "   THE ROOTS ARE:"
216  PRINT : PRINT "ROOT #   REAL PART      IMAGINARY PART"
217  FOR I = 1 TO 0
218  RR(I) =  INT (RR(I) * 1E6 + .5) / 1E6:RI(I) =  INT (RI(I) * 1E6 + .5)
     / 1E6
219  PRINT I; TAB( 10)RR(I); TAB( 25)RI(I): NEXT I
220  PRINT : PRINT
221  END
```

Fig. 5-1 (cont). Listing of PLYROOTS program.

```
ORDER OF POLYNOMIAL? 2

ENTER COEFFICIENTS STARTING WITH
THE CONSTANT TERM

COEFFICIENT A(0) ?-6
COEFFICIENT A(1) ?1
COEFFICIENT A(2) ?1

COMPUTING

FOR THE POLYNOMIAL EQUATION:

0 = -6 + 1X     + 1X^2

   THE ROOTS ARE:

ROOT #   REAL PART     IMAGINARY PART
1        2             0
2        -3            0
```

Fig. 5-2. Results of determining the roots for the second-order polynomial.

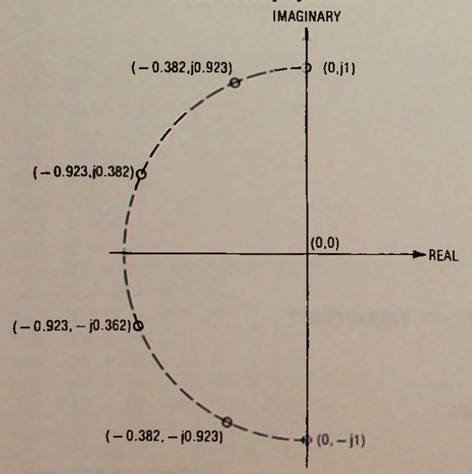


Fig. 5-4. Pole locations for fourth-order Butterworth polynomial.

$$F(s) = \frac{a_0 + a_1 s + a_2 s^2 + \ldots}{(s + b_1)(s + b_2) + \ldots} \quad \text{(Eq. 5-3)}$$

The polynomial coefficients a_i of the numerator are assumed to be real, while the denominator has been factored. Consequently the poles (i.e., the roots) of the denominator of the transfer function are known, and can be either real or complex. If this is not the case, the PLYROOT program discussed earlier in this chapter solves for both real and complex roots of a polynomial.

When executed, the program requires the number of numerator coefficients and their values, and the number of roots. Each denominator root

```
ORDER OF POLYNOMIAL? 4

ENTER COEFFICIENTS STARTING WITH
THE CONSTANT TERM

COEFFICIENT A(0) ?1
COEFFICIENT A(1) ?2.613
COEFFICIENT A(2) ?3.414
COEFFICIENT A(3) ?2.613
COEFFICIENT A(4) ?1

COMPUTING

FOR THE POLYNOMIAL EQUATION:

0 = 1 + 2.613X +   3.414X^2
    + 2.613X^3
    + 1X^4

   THE ROOTS ARE:

ROOT #   REAL PART       IMAGINARY PART
1        -.382629        -.923902
2        -.382629        .923902
3        -.923871        -.382705
4        -.923871        .382705
```

Fig. 5-3. Results of determining pole locations for the fourth-order Butterworth polynomial.


```
100  REM   INVERSE LAPLACE TRANSFORM (ILAPLACE)
101  CLEAR : HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "INVERSE LA
     PLACE TRANSFORM": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD.":
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR I = 1 TO 2500: NEXT : HOME
105  DIM A(20),CI(20),CR(20),RI(20),SR(20),SI(20),DR(20),DI(20)
106  INPUT "NUMBER OF NUMERATOR COEFFICIENTS? ";M: PRINT
107  FOR I = 1 TO M
108  PRINT "COEFFICIENT ";I;
109  INPUT "? ";A(I): NEXT I
110  PRINT : INPUT "NUMBER OF DENOMINATOR ROOTS? ";N: PRINT
111  FOR I = 1 TO N
112  PRINT "ROOT ";I;" (REAL,IMAGINARY): ";
113  INPUT "? ";RR(I),RI(I)
114  NEXT I
115  FOR I = 1 TO N
116  S1 = RR(I)
117  S2 = RI(I)
118  PR = 1
119  PI = 0
120  TR = A(1)
121  TI = 0
122  IF (M - 1) < 0 THEN 163
123  IF (M - 1) <  = 0 THEN 132
124  FOR J = 2 TO M
125  B1 = PR * S1 - PI * S2
126  C1 = PI * S1 + PR * S2
127  PR = B1
128  PI = C1
129  TR = TR + A(J) * PR
130  TI = TI + A(J) * PI
131  NEXT J
132  SR(I) = TR
133  SI(I) = TI
134  NEXT I
135  FOR J = 1 TO N
136  PR = 1
137  PI = 0
138  FOR I = 1 TO N
139  IF J - I = 0 THEN 146
140  UR = RR(J) - RR(I)
141  UI = RI(J) - RI(I)
142  B1 = PR * UR - PI * UI
143  C1 = PR * UI + PI * UR
144  PR = B1
145  PI = C1
146  NEXT I
147  DR(J) = PR
148  DI(J) = PI
149  NEXT J
150  FOR I = 1 TO N
151  CR(I) = (SR(I) * DR(I) + SI(I) * DI(I)) / ((DR(I) ^ 2) + (DI(I) ^ 2))
152  CI(I) = (SI(I) * DR(I) - SR(I) * DI(I)) / (DR(I) * DR(I) + DI(I) * DI
     (I))
153  NEXT I
```

Continued on next page.

Fig. 5-5. Listing for ILAPLACE program.

```
...RAM,": PRINT "AS A FUNCTION
...<";CR(I);" + J(";CI(I);")> * EXP<"
...(I);")>*T"
...R I = 2 TO N
160  PRINT : PRINT " + <";CR(I);" + J(";CI(I);")> * EXP<";RR(I);
161  PRINT " + J(";RI(I);")>*T"
162  NEXT I
163  END
```

Fig. 5-5 (cont). Listing for ILAPLACE program.

is assumed to be a complex number and is entered in the format A,B where A and B are the real and imaginary parts respectively. In addition, all roots must be distinct so that multiple roots are not allowed. The computed inverse Laplace transform time response is expressed in the form:

$$f(t) = (a_1 + jb_1)e^{(c_1 + jd_1)t} + (a_2 + jb_2)e^{(c_2 + jd_2)t} + \ldots \qquad \text{(Eq. 5-4)}$$

so that there will be as many terms as there are roots.

Example 5-3

Determine the inverse Laplace transform time response for the transfer function:

$$T(s) = \frac{s-1}{(s+2)(s+1)}$$

There are two coefficients for the numerator and two real denominator roots. Fig. 5-6 shows the

```
NUMBER OF NUMERATOR COEFFICIENTS? 2

COEFFICIENT 1? -1
COEFFICIENT 2? 1

NUMBER OF DENOMINATOR ROOTS? 2

ROOT 1 (REAL, IMAGINARY): ? -2, 0
ROOT 2 (REAL, IMAGINARY): ? -1, 0

THE INVERSE LAPLACE TRANSFORM,
AS A FUNCTION OF TIME IS:

F(T) =

    <3 + J(0)> * EXP<-2 + J(0)>*T

  + <-2 + J(0)> * EXP<-1 + J(0)>*T
```

Fig. 5-6. Results for inverse LAPLACE transfer function (Example 5-3).

```
NUMBER OF NUMERATOR COEFFICIENTS? 1

COEFFICIENT 1? 1

NUMBER OF DENOMINATOR ROOTS? 2

ROOT 1 (REAL, IMAGINARY): ? -2, -1
ROOT 2 (REAL, IMAGINARY): ? -2, 1

THE INVERSE LAPLACE TRANSFORM,
AS A FUNCTION OF TIME IS:

F(T) =

    <0 + J(.5)> * EXP<-2 + J(-1)>*T

  + <0 + J(-.5)> * EXP<-2 + J(1)>*T
```

Fig. 5-7. Results for inverse LAPLACE transfer function (Example 5-4).

output results for this transfer function which has the time variant solution:

$$f(t) = 3e^{-2t} - 2e^{-t}$$

which agrees with the result obtained by the use of inverse Laplace transform tables.

Example 5-4

Determine the inverse Laplace transform time response for the transfer function:

$$T(s) = \frac{1}{s^2 + 4s + 5} = \frac{1}{(s+2+j)(s+2-j)}$$

For this transfer function, there are 2 complex roots which are complex conjugates. The output results are shown in Fig. 5-7, but can be simplified further.

The computed solution can then be rewritten as:

$$\begin{aligned} &j0.5e^{.5t}e^{-(2+j)t} - j0.5e^{-(2-j)t} \\ &\quad = e^{-2t}(0.5je^{-t} - 0.5je^{t}) \\ &\quad = e^{-2t}\sin(t) \end{aligned} \qquad \text{(Eq. 5-5)}$$


```
NUMBER OF NUMERATOR COEFFICIENTS? 1

COEFFICIENT 1? 1

NUMBER OF DENOMINATOR ROOTS? 3

ROOT 1 (REAL, IMAGINARY): ? 0, 0
ROOT 2 (REAL, IMAGINARY): ? -1, 0
ROOT 3 (REAL, IMAGINARY): ? 1, 0

THE INVERSE LAPLACE TRANSFORM,
AS A FUNCTION OF TIME IS:

F(T) =

    <-1 + J(0)> * EXP<0 + J(0)>*T

  + <.5 + J(0)> * EXP<-1 + J(0)>*T

  + <.5 + J(0)> * EXP<1 + J(0)>*T
```

Fig. 5-8. Results for inverse LAPLACE transfer function (Example 5-5).

using the exponential identities of trigonometric functions.

Example 5-5

Determine the inverse Laplace transform time response for the transfer function:

$$T(s) = \frac{1}{s(s^2-1)} = \frac{1}{s(s+1)(s-1)}$$

Fig. 5-8 shows the output results, which can be simplified further. Using the identity:

$$\frac{e^t + e^{-t}}{2} = \cosh(t) \qquad \text{(Eq. 5-6)}$$

so that the computed answer can now be more formally stated as:

$$f(t) = \cosh(t) - 1$$

which can be verified using tables of inverse Laplace transforms.

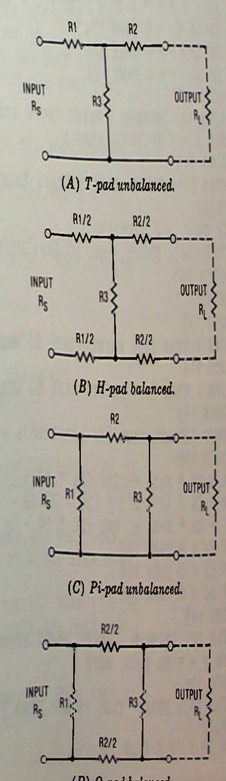
IMPEDANCE MATCHING PADS

Impedance matching devices are frequently used to "match" the signal source to a given load impedance. These devices may be either transformers, transistor circuits, or a class of networks called pads. Pads may be composed solely of resistors, or they may be a combination of inductors and capacitors. In this section, programs for the design of several common types of resistive and lossless matching pads are discussed.

Resistance Pads

A resistance pad is a 4-terminal resistor network that permits the matching of a signal source to a load, but with a finite amount of loss, or attenuation. This is because resistors are passive devices and, unlike a transistor circuit or amplifier, cannot produce gain. The four main types of resistance pad networks are shown in Fig. 5-9. The T- and Pi-networks are unbalanced pads while the H- and O-pads are their respective balanced counterparts. When the source and load impedances are equal, these networks are referred to as attenuators, otherwise they are called pads when the source and load impedances are not equal.

The ATTNPAD program shown in Fig. 5-10 is a graphics program to determine the required pad resistances when the source and load impedances, as well as the required amount of attenuation (in decibels), are known. The final schematic is included as part of the video display.



(A) *T-pad unbalanced.*

(B) *H-pad balanced.*

(C) *Pi-pad unbalanced.*

(D) *O-pad balanced.*

Fig. 5-9. Resistive matching pads.

```
100  REM   ATTENUATOR PAD DESIGN WITH GRAPHICS (ATTNPAD)
101  HOME : VTAB (7): PRINT  TAB( 9): INVERSE : PRINT "ATTENUATOR PAD DES
     IGN": NORMAL : PRINT : PRINT  TAB( 13): INVERSE : PRINT "WITH GRAPHI
     CS": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II
105  HOME : PRINT "ATTENUATOR DESIGN:"
106  CLEAR
107  PRINT : PRINT "   1. T-PAD (UNBALANCED)"
108  PRINT : PRINT "   2. H-PAD (BALANCED)"
109  PRINT : PRINT "   3. PI-PAD (UNBALANCED)"
110  PRINT : PRINT "   4. O-PAD (BALANCED)"
111  POKE  - 16368,0: PRINT : PRINT "   5. QUIT": PRINT : INPUT "SELECT #
     ? ";Z9: IF Z9 < 1 OR Z9 > 5 THEN 105
112  ON Z9 GOTO 113,133,135,155,999
113  HOME : VTAB (2): PRINT  TAB( 9)"UNBALANCED T-PAD DESIGN"
114  GOSUB 163
115  INPUT "ATTENUATION (IN DECIBELS): ?";N
116  GOSUB 158
117  ZA = (ZS * C - 2 * X * D) / Z9:ZA =  INT (ZA * 10 + .5) / 10
118  ZB = (ZL * C - 2 * X * D) / Z9:ZB =  INT (ZB * 10 + .5) / 10
119  ZC = 2 * X * D:ZC =  INT (ZC * 10 + .5) / 10
120  IF ZA <  = 0 THEN 124
121  IF ZB <  = 0 THEN 124
122  IF ZC <  = 0 THEN 124
123  GOTO 127
124  VTAB 12: INVERSE : PRINT "MORE ATTENUATION IS REQUIRED, TRY AGAIN": NORMAL
125  FOR F = 1 TO 1400: NEXT F
126  GOTO 113
127  VTAB (5): PRINT  TAB( 10)ZA;: PRINT  TAB( 28)ZB: VTAB (12): PRINT  TAB(
     25)ZC
128  IF Z9 <  > 2 THEN 130
129  VTAB (14): PRINT  TAB( 10)ZA;: PRINT  TAB( 28)ZB
130  GOSUB 168
131  PRINT " "
132  GOTO 157
133  HOME : VTAB (2): PRINT  TAB( 9)"BALANCED H-PAD DESIGN"
134  GOTO 114
135  HOME : VTAB (2): PRINT  TAB( 8)"UNBALANCED PI-PAD DESIGN"
136  GOSUB 163
137  INPUT "ATTENUATION (IN DECIBELS): ?";N
138  GOSUB 158
139  ZA = ZS * E:ZA =  INT (ZA * 10 + .5) / 10
140  ZB = (X / (D * 2)) / (Z9 - 2):ZB =  INT (ZB * 10 + .5) / 10
141  ZC = ZL * E:ZC =  INT (ZC * 10 + .5) / 10
142  IF ZA < 0 THEN 146
143  IF ZB < 0 THEN 146
144  IF ZC < 0 THEN 146
145  GOTO 149
146  VTAB 12: INVERSE : PRINT "MORE ATTENUATION REQUIRED, TRY AGAIN": NORMAL
147  FOR F = 1 TO 1400: NEXT F
148  GOTO 135
149  VTAB (5): PRINT  TAB( 19)ZB: VTAB (15): PRINT  TAB( 7)ZA;: PRINT  TAB(
     28)ZC
150  IF Z9 <  > 4 THEN 152
```

Continued on next page.

Fig. 5-10. Listing for ATTNPAD program.


```
151  VTAB (19): PRINT  TAB( 19)ZB
152  GOSUB 181
153  PRINT " "
154  GOTO 157
155  HOME : VTAB (2): PRINT  TAB( 9)"BALANCED O-PAD DESIGN"
156  GOTO 136
157  VTAB 24: POKE  - 16368,0: CALL  - 756: GOTO 105: REM          BACK
     TO BEGINNING
158  K2 = 10 ^ (N / 10):K =  SQR (K2)
159  C = (K2 + 1) / (K2 - 1):D = K / (K2 - 1)
160  Z =  SQR (ZS / ZL):X =  SQR (ZL * ZS)
161  E = (K2 - 1) / (K2 - (2 * K / Z) + 1)
162  RETURN
163  VTAB (20): PRINT "INPUT IMPEDANCE: ";
164  INPUT ZS
165  PRINT "LOAD IMPEDANCE: ";
166  INPUT ZL: RETURN
167  VTAB (23): END
168  VTAB (12): PRINT "INPUT": VTAB (13): PRINT  TAB( 35)"LOAD"
169  POKE 1805,32: POKE 1806,32: POKE 1807,32: POKE 1808,32: POKE 1816,32
     : POKE 1817,32: POKE 1818,32: POKE 1819,32: POKE 1930,32: POKE 1931,
     32: POKE 1932,32: POKE 1933,32
170  FOR I = 1936 TO 1944: POKE I,32: NEXT I
171  POKE 1947,32: POKE 1948,32: POKE 1949,32: POKE 1950,32: POKE 1077,32
     : POKE 1078,32: POKE 1079,32: POKE 1080,32: POKE 1084,32: POKE 1088,
     32: POKE 1089,32: POKE 90,32: POKE 1090,32: POKE 1091,32
172  POKE 1212,32: POKE 1339,32: POKE 1340,32: POKE 1341,32: POKE 1467,32
     : POKE 1469,32: POKE 1595,32: POKE 1597,32: POKE 1723,32: POKE 1724,
     32: POKE 1725,32: POKE 1852,32: POKE 1980,32
173  IF Z9 <  > 2 THEN 179
174  POKE 1973,32: POKE 1974,32: POKE 1975,32: POKE 1976,32: POKE 1984,32
     : POKE 1985,32: POKE 1986,32: POKE 1987,32
175  POKE 1114,32: POKE 1115,32: POKE 1116,32: POKE 1117,32: FOR I = 1120
      TO 1128: POKE I,32: NEXT I
176  POKE 1131,32: POKE 1132,32: POKE 1133,32: POKE 1134,32
177  POKE 1245,32: POKE 1246,32: POKE 1247,32: POKE 1248,32: POKE 1256,32
     : POKE 1257,32: POKE 1258,32: POKE 1259,32
178  RETURN
179  FOR I = 1114 TO 1134: POKE I,32: NEXT I
180  RETURN
181  VTAB (12): PRINT "INPUT": VTAB (13): PRINT  TAB( 35)"LOAD"
182  POKE 1810,32: POKE 1811,32: POKE 1812,32: POKE 1813,32
183  FOR I = 1931 TO 1938: POKE I,32: NEXT I
184  FOR I = 1941 TO 1949: POKE I,32: NEXT I
185  POKE 1079,32: POKE 1082,32: POKE 1083,32: POKE 1084,32: POKE 1085,32
     : POKE 1088,32: POKE 1207,32: POKE 1216,32: POKE 1334,32: POKE 1335,
     32: POKE 1336,32
186  POKE 1343,32: POKE 1344,32: POKE 1345,32: POKE 1462,32: POKE 1464,32
     : POKE 1471,32: POKE 1473,32: POKE 1590,32: POKE 1592,32: POKE 1599,
     32: POKE 1601,32
187  POKE 1718,32: POKE 1719,32: POKE 1720,32: POKE 1727,32: POKE 1728,32
     : POKE 1729,32: POKE 1847,32: POKE 1856,32: POKE 1975,32: POKE 1984,
     32
188  IF Z9 <  > 4 THEN 193
189  POKE 1975,32: POKE 1978,32: POKE 1979,32: POKE 1980,32: POKE 1981,32
     : POKE 1984,32: FOR I = 1114 TO 1122: POKE I,32: NEXT I
190  FOR I = 1125 TO 1132: POKE I,32: NEXT I
191  FOR I = 1250 TO 1253: POKE I,32: NEXT I
```

Continued on next page.

Fig. 5-10 (cont). Listing for ATTNPAD program.

```
192  RETURN
193  FOR I = 1114 TO 1134: POKE I,32: NEXT I
194  RETURN
999  HOME : VTAB 12: PRINT "GOODBYE....."
```

Fig. 5-10 (cont). Listing for ATTNPAD program.

Depending on the relationship between the source and load impedances, there is usually a minimum value of attenuation required in order to satisfy the design. If more attenuation is required, the program informs us of that fact.

Example 5-6

Design a T-pad having an attenuation of 10 dB while matching a 300-ohm source to a 75-ohm load. When run, we find that it is not possible to have 10-dB attenuation. Consequently, we must choose a higher value, such as 15 dB. Fig. 5-11 shows the final circuit required.

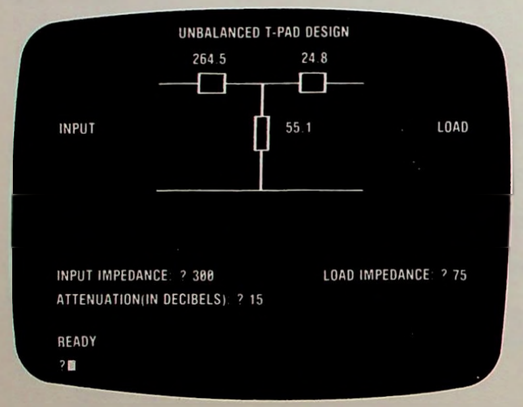


Fig. 5-11. Video results of T-pad having an attenuation of 15 dB.

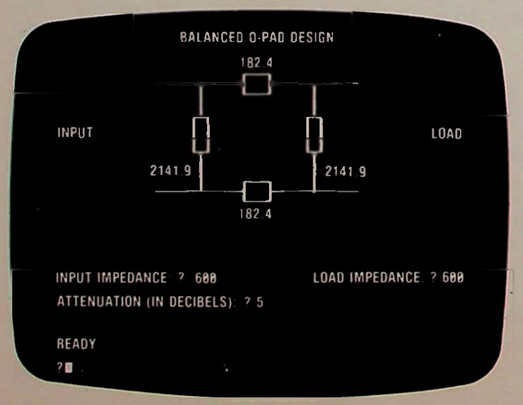


Fig. 5-12. Video results of O-pad having an attenuation of 5 dB.

Example 5-7

Design an O-pad having an attenuation of 5 dB between a 600-ohm source and load. Fig. 5-12 shows the final circuit required.

Lossless Pads

Fig. 5-13 shows two types of lossless pads that are used to match impedances at a single frequency. Both circuits are equivalent, and require that the source impedance be less than the load impedance. If the source impedance is greater than the load, then either circuit is turned around so that the shunt element is now in parallel with the source instead of with the load.

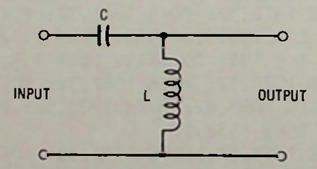
Fig. 5-14 lists the LCPAD program which determines the required series capacitor/shunt inductor circuit (Fig. 5-13A), or the equivalent series inductor/shunt capacitor circuit (Fig. 5-13B) when the source and load impedances, as well as the input frequency, are known. The LC-PAD program includes a schematic which is included as part of the video display.

Example 5-8

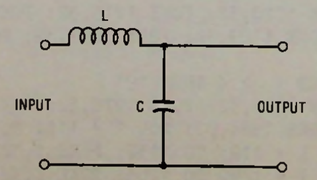
Design a lossless pad to match a 50-ohm source to a 75-ohm load at 3.5 MHz.

The two possible circuits are shown in Fig. 5-15, along with the required values.

Another version of the lossless pad is to be able to specify the phase shift in electrical degrees.



(A) Series capacitor/shunt inductor.



(B) Series inductor/shunt capacitor.

Fig. 5-13. LC lossless pads for a specific frequency.


```
100  REM   LOSSLESS L-C PAD DESIGN - GRAPHICS (LCPAD)
101  HOME : VTAB (7): PRINT  TAB( 9): INVERSE : PRINT "LOSSLESS L-C PAD D
     ESIGN": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II
105  HOME : PRINT : PRINT  TAB( 10)"LOSSLESS L-C PAD DESIGN"
106  VTAB (21): PRINT "INPUT IMPEDANCE ";
107  INPUT Z1
108  PRINT "LOAD IMPEDANCE ";
109  INPUT Z2: IF Z2 < Z1 THEN 105
110  INPUT "INPUT FREQUENCY (IN HZ)? ";F
111  W = F / 6.28318
112  X1 = Z2 *  SQR (Z1 / (Z2 - Z1)):X2 =  SQR (Z1 * (Z2 - Z1))
113  S1 = 1 * 1E9 / (W * X2):H1 = X1 * 1E6 / W
114  S1 =  INT (S1 * 100 + .5) / 100:H1 =  INT (H1 * 100 + .5) / 100
115  S2 = X2 * 1E6 / W:H2 = 1 * 1E9 / (W * X1)
116  S2 =  INT (S2 * 100 + .5) / 100:H2 =  INT (H2 * 100 + .5) / 100
117  VTAB (4): PRINT  TAB( 11)S1;" NF"; TAB( 32)S2;" UH"
118  VTAB (15): PRINT  TAB( 13)H1;" UH": VTAB (13): PRINT  TAB( 25)H2;" N
     F"
119  VTAB (10): PRINT  TAB( 20)"OR";
120  FOR I = 1 TO 1000: READ X: IF X = 999 GOTO 130
121  POKE X,32: NEXT I
122  DATA  1540,1542,1562,1563,1564,1565,1566,1567,1568
123  DATA  1665,1666,1667,1668,1670,1671,1672,1673,1674,1675,1676,1677,16
     87,1688,1689,1690,1692,1694,1696,1697,1698,1699,1700,1701,1702
124  DATA  1796,1798,1801,1820,1822,1826,1929,1930,1954
125  DATA  1074,1098,1200,1201,1202,1225,1226,1227,1330
126  DATA  1456,1457,1458,1481,1482,1483,1586,1610
127  DATA  1713,1714,1738,1841,1866,1969,1994
128  DATA  1104,1105,1106,1107,1108,1109,1110,1111,1112,1113,1114,1115,11
     16,1117,1127,1128,1129,1130,1131,1132,1133,1134,1135,1136,1137,1138,
     1139,1140,1141,1142
129  DATA  999
130  VTAB (23): END
```

Fig. 5-14. Listing for LCPAD program.

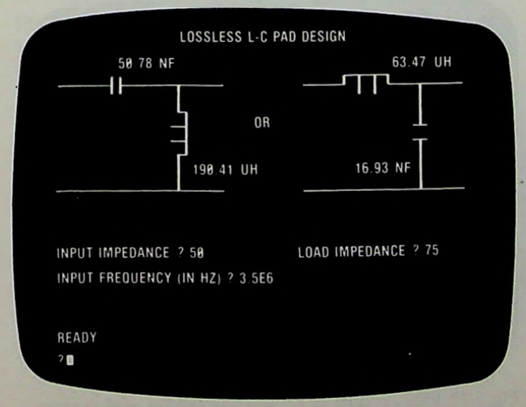


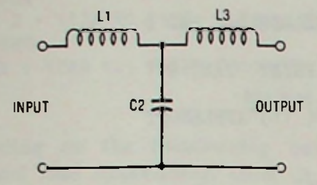
Fig. 5-15. Video results of lossless LC pad design.

Depending on the source and load impedances, and the amount of phase shift required, there are four circuits possible, as shown in Fig. 5-16. The convention is taken that if the input signal is to lead the output, the phase angle is positive; otherwise, it is negative if the input lags the output.

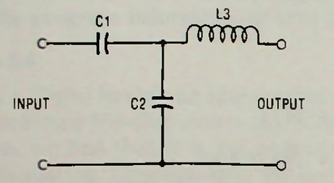
The LCPSPAD program of Fig. 5-17 is used for designing these types of lossless pads. Like ATTNPAD and LCPAD, this program includes the final schematic with component values as part of the video display.

Example 5-9

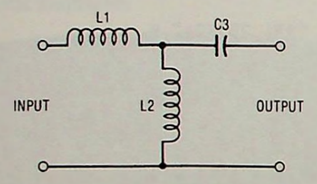
Design a lossless pad to match a 50-ohm source with a 75-ohm load with a phase shift of +45 degrees at 3.5 MHz. Repeat this example, but for a phase shift of −45 degrees.



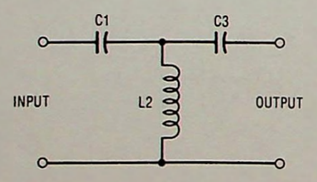
(A) Large lag section.



(B) Small lag section.



(C) Large lead section.



(D) Small lead section.

Fig. 5-16. LC lossless pads for a specific phase shift.

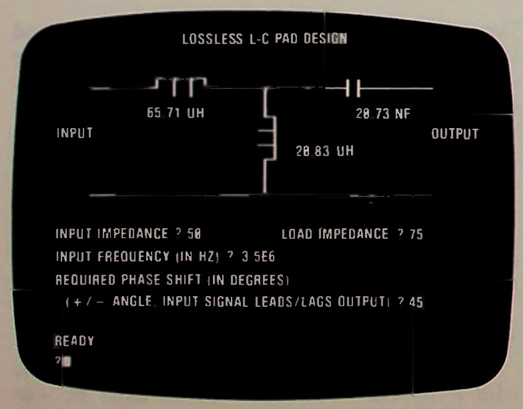


Fig. 5-18. Video results of LC lossless pad for +45 degrees phase shift.

The final circuit for +45 degrees phase shift is shown in Fig. 5-18, while the display of Fig. 5-19 shows the required circuit having a phase shift of −45 degrees.

PI-TEE (DELTA-WYE) TRANSFORMS

Networks often topologically appear in forms resembling either "Tee" or "Pi" shaped networks (sometimes called Wye or Delta networks), as shown in Fig. 5-20. Examples are the resistive attenuators discussed previously, Wheatstone bridge, and lattice circuits. The ability to transform from Tee to Pi, or vice versa, simplifies the analysis of a number of circuits.

The TEEPI program listed in Fig. 5-21 allows the user to determine the corresponding equivalent network. The starting and equivalent schematics are displayed in the screen along with their values.

Example 5-10

For the unbalanced T-pad design of Example 5-6, determine the equivalent Pi (or Delta) circuit.

From the results of Example 5-6, Z1, Z2, and Z3 are 264.5, 55.1, and 24.8 ohms respectively. The equivalent Pi circuit is shown in Fig. 5-22. In fact, the resulting Pi circuit should be the same as if we had chosen to design an unbalanced Pi-pad in Example 5-6 instead of a T-pad.

MESH CURRENT NETWORK ANALYSIS

In an electric network, mesh currents (or loop currents) are produced in each circuit branch as a result of one or more applied voltage sources. From these mesh currents, one can then determine

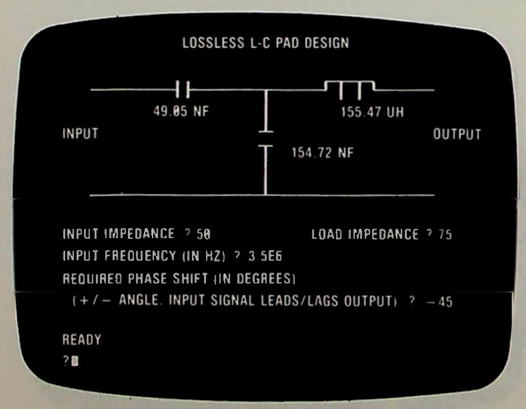


Fig. 5-19. Video results of LC lossless pad for −45 degrees phase shift.


```
100  REM  LOSSLESS L-C PAD WITH PHASE SHIFT (WITH GRAPHICS)  LCPSPAD
101  HOME : VTAB (7): PRINT  TAB( 8): INVERSE : PRINT "LOSSLESS L-C PAD D
     ESIGN": NORMAL : PRINT : PRINT  TAB( 12): INVERSE : PRINT "WITH PHAS
     E SHIFT": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II
105  HOME : VTAB (2): PRINT  TAB( 7)"LOSSLESS L-C PAD DESIGN"
106  VTAB (17): PRINT "INPUT IMPEDANCE?";
107  INPUT " ";Z1
108  PRINT "LOAD IMPEDANCE?";
109  INPUT " ";Z2
110  INPUT "INPUT FREQUENCY (IN HZ)? ";F
111  PRINT "REQUIRED PHASE SHIFT IN DEGREES "
112  PRINT " (+/- ANGLE": INPUT "SIGNAL LEADS/LAGS OUTPUT)?";D
113  W = F / 6.28319:R = D / 57.2957
114  Z = Z1 / Z2:Y =  SQR (Z)
115  A = Z *  SIN (R) / (Y -  COS (R))
116  B = Y *  SIN (R)
117  C = Y *  SIN (R) / (1 - Y *  COS (R))
118  IF R < 0 THEN 120
119  IF R > 0 THEN 137: GOTO 118
120  X3 = Z1 /  ABS (B):L3 = X3 * 1E6 / W
121  X2 = Z1 /  ABS (A):C2 = 1E9 / (W * X2)
122  L3 =  INT (L3 * 100 + .5) / 100
123  C2 =  INT (C2 * 100 + .5) / 100
124  IF C < 0 THEN 126
125  IF C > 0 THEN 132: GOTO 124
126  X1 = Z1 /  ABS (C):C1 = 1E9 / (W * X1)
127  C1 =  INT (C1 * 100 + .5) / 100
128  L3 =  INT (L3 * 100 + .5) / 100
129  VTAB (8): PRINT  TAB( 6)C1;" NF";: PRINT  TAB( 25)L3;" UH"
130  VTAB (10): PRINT " INPUT";: PRINT  TAB( 33)"OUTPUT": VTAB (12): PRINT
     TAB( 23)C2;" NF"
131  GOSUB 153: GOTO 152
132  X1 = Z1 / C:L1 = X1 * 1E6 / W
133  L1 =  INT (L1 * 100 + .5) / 100
134  VTAB (8): PRINT  TAB( 6)L1;" UH";: PRINT  TAB( 25)L3;" UH"
135  VTAB (10): PRINT " INPUT";: PRINT  TAB( 33)"OUTPUT": VTAB (12): PRINT
     TAB( 23)C2;" NF"
136  GOSUB 161: GOTO 152
137  X2 = Z1 / A:L2 = X2 * 1E6 / W:X3 = Z1 / B:C3 = 1E9 / (W * X3)
138  L2 =  INT (L2 * 100 + .5) / 100
139  C3 =  INT (C3 * 100 + .5) / 100
140  IF C < 0 THEN 142
141  IF C > 0 THEN 147: GOTO 140
142  X1 = Z1 /  ABS (C):C1 = 1E9 / (W * X1)
143  C1 =  INT (C1 * 100 + .5) / 100
144  VTAB (8): PRINT  TAB( 6)C1;" NF";: PRINT  TAB( 25)C3;" NF"
145  VTAB (10): PRINT "INPUT";: PRINT  TAB( 33)"OUTPUT": VTAB (12): PRINT
     TAB( 23)L2;" UH"
146  GOSUB 169: GOTO 152
147  X1 = Z1 / C:L1 = X1 * 1E6 / W
148  L1 =  INT (L1 * 100 + .5) / 100
149  VTAB (8): PRINT  TAB( 6)L1;" UH";: PRINT  TAB( 25)C3;" NF"
```

Continued on next page.

Fig. 5-17. Listing for LCPSPAD program.

```
150  VTAB (10): PRINT "INPUT";: PRINT  TAB( 33)"OUTPUT": VTAB (12): PRINT
     TAB( 23)L2;" UH"
151  GOSUB 177
152  VTAB (23): END
153  POKE 1417,32: POKE 1419,32: FOR I = 1432 TO 1438: POKE I,32: NEXT I
154  FOR I = 1540 TO 1545: POKE I,32: NEXT I
155  FOR I = 1547 TO 1560: POKE I,32: NEXT I
156  POKE 1562,32: POKE 1564,32: FOR I = 1566 TO 1570: POKE I,32: NEXT I
157  POKE 1673,32: POKE 1675,32: POKE 1682,32: POKE 1690,32: POKE 1692,32
     : POKE 1810,32: POKE 1818,32: POKE 1820,32: POKE 1938,32: POKE 1081,
     32: POKE 1082,32: POKE 1083,32
158  POKE 1337,32: POKE 1338,32: POKE 1339,32: POKE 1466,32: POKE 1594,32
     : POKE 1722,32
159  FOR I = 1836 TO 1866: POKE I,32: NEXT I
160  RETURN
161  FOR I = 1415 TO 1421: POKE I,32: NEXT I: FOR I = 1432 TO 1438: POKE
     I,32: NEXT I
162  FOR I = 1540 TO 1543: POKE I,32: NEXT I: POKE 1545,32: POKE 1547,32
163  FOR I = 1549 TO 1560: POKE I,32: NEXT I: POKE 1562,32: POKE 1564,32:
      FOR I = 1566 TO 1569: POKE I,32: NEXT I
164  POKE 1673,32: POKE 1675,32: POKE 1683,32: POKE 1690,32: POKE 1692,32
165  POKE 1811,32: POKE 1939,32: POKE 1082,32: POKE 1083,32: POKE 1084,32
     : POKE 1338,32: POKE 1339,32: POKE 1340,32
166  POKE 1467,32: POKE 1595,32: POKE 1723,32
167  FOR I = 1835 TO 1865: POKE I,32: NEXT I
168  RETURN
169  POKE 1417,32: POKE 1419,32: POKE 1434,32: POKE 1436,32: FOR I = 1539
      TO 1545: POKE I,32: NEXT I
170  FOR I = 1547 TO 1562: POKE I,32: NEXT I
171  FOR I = 1564 TO 1568: POKE I,32: NEXT I
172  POKE 1673,32: POKE 1675,32: POKE 1683,32: POKE 1690,32: POKE 1692,32
     : POKE 1811,32: POKE 1812,32: POKE 1940,32: POKE 1081,32: POKE 1082,
     32: POKE 1083,32: POKE 1084,32
173  POKE 1212,32: FOR I = 1337 TO 1340: POKE I,32: NEXT I
174  POKE 1468,32: POKE 1595,32: POKE 1596,32: POKE 1723,32
175  FOR I = 1835 TO 1865: POKE I,32: NEXT I
176  RETURN
177  FOR I = 1415 TO 1421: POKE I,32: NEXT I
178  POKE 1434,32: POKE 1436,32: POKE 1540,32: POKE 1541,32: POKE 1542,32
     : POKE 1543,32: POKE 1545,32: POKE 1547,32
179  FOR I = 1549 TO 1562: POKE I,32: NEXT I: FOR I = 1564 TO 1569: POKE
     I,32: NEXT I
180  POKE 1673,32: POKE 1675,32: POKE 1683,32: POKE 1690,32: POKE 1692,32
     : POKE 1811,32: POKE 1812,32
181  POKE 1940,32: POKE 1081,32: POKE 1082,32: POKE 1083,32: POKE 1084,32
     : POKE 1212,32: FOR I = 1337 TO 1340: POKE I,32: NEXT I
182  POKE 1468,32: POKE 1595,32: POKE 1596,32: POKE 1723,32
183  FOR I = 1835 TO 1865: POKE I,32: NEXT I
184  RETURN
```

Fig. 5-17 (cont). Listing for LCPSPAD program.

the voltage across or the current through any network element.

Once the mesh equations are established, the individual mesh currents are found by solving a set of simultaneous equations. Fig. 5-23 shows the SIMULEQU program listing to solve a set of simultaneous equations having real coefficients if a solution is possible. When run, the program requires the number of equations, which is the same number of unknown variables and the minimum allowable value for any solution, which usually is set at 1E-6 (10^{-6}). The coefficients are then entered, one at a time for each equation, with the constant term entered last. For the equation: $9X - 8Y + 3Z = 5$, the "9" is the first coefficient, while the "5" is the fourth coefficient.

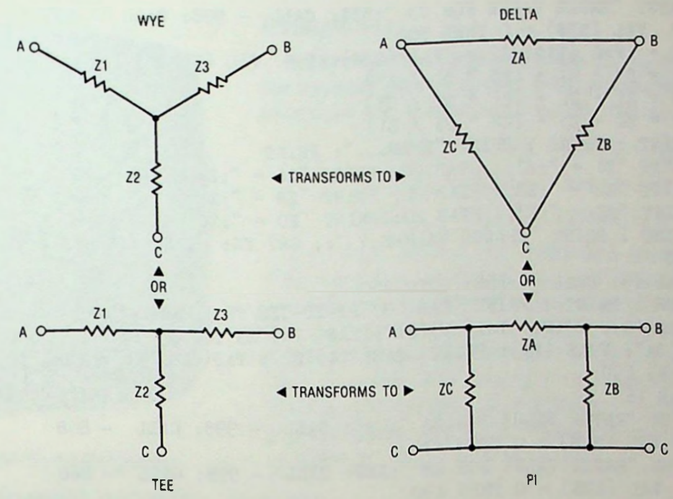


Fig. 5-20. Pi-Tee and Delta-Wye equivalent networks.

```
100  REM  PI-TEE (DELTA-WYE) TRANSFORM -(TEEPI)
110  HOME
120  VTAB (7): PRINT  TAB( 11): INVERSE : PRINT "PI-TEE (DELTA-WYE)": NORMAL
130  PRINT : PRINT  TAB( 13): INVERSE : PRINT "TRANSFORMATION": NORMAL : PRINT
     : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"HOWARD
     M. BERLIN, LTD."
140  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
150  FOR II = 1 TO 2500: NEXT II: HOME
160  PRINT : PRINT "SELECT:"
170  PRINT "       (T)EE-PI"
180  PRINT : PRINT "       (P)I-TEE"
190  PRINT : INPUT "INPUT CHOICE (T OR P)? ";C$
200  IF C$ <  > "T" AND C$ <  > "P" THEN 190
210  IF C$ = "P" THEN 430
220  HOME : PRINT : PRINT  TAB( 8)"TEE-TO-PI TRANSFORMATION"
230  VTAB (8): PRINT  TAB( 5)"Z1"; TAB( 16)"Z3"; TAB( 19)"TO-->"; TAB( 29
     )"ZA": VTAB (11): PRINT  TAB( 14)"Z2"; TAB( 22)"ZC"; TAB( 38)"ZB"
240  GOSUB 650
250  VTAB (16)
260  INPUT "ENTER VALUE FOR Z1 ";Z1$: CALL  - 998: CALL  - 868
270  IF  VAL (Z1$) = 0 THEN 260
280  INPUT "ENTER VALUE FOR Z2 ";Z2$: CALL  - 998: CALL  - 868
290  IF  VAL (Z2$) = 0 THEN 280
```

Continued on next page.

Fig. 5-21. Listing for TEEPI program.

```
300  INPUT "ENTER VALUE FOR Z3 ";Z3$: CALL  - 998: CALL  - 868
310  IF  VAL (Z3$) = 0 THEN 300
320  Z1 =  VAL (Z1$):Z2 =  VAL (Z2$):Z3 =  VAL (Z3$)
340  ZA = Z3 + Z1 + (Z3 * Z1 / Z2)
350  ZB = Z1 + Z2 + (Z1 * Z2 / Z3)
360  ZC = Z2 + Z3 + (Z2 * Z3 / Z1)
370  PRINT : PRINT : PRINT "THEN...": PRINT
380  PRINT "Z1 = ";Z1;: HTAB 20: PRINT "ZA = ";ZA
390  PRINT "Z2 = ";Z2;: HTAB 20: PRINT "ZB = ";ZB
400  PRINT "Z3 = ";Z3;: HTAB 20: PRINT "ZC = ";ZC
410  PRINT : PRINT "CHANGE VALUES ? ";: GET Y$: HTAB 1: IF Y$ <  > "Y" THEN
     800
420  VTAB 16: CALL  - 868: GOTO 250
430  HOME : PRINT : PRINT  TAB( 9)"PI-TO-TEE TRANSFORMATION"
440  VTAB (8): PRINT  TAB( 5)"Z1"; TAB( 16)"Z3"; TAB( 18)" <--TO"; TAB( 2
     9)"ZA": VTAB (11): PRINT  TAB( 14)"Z2"; TAB( 22)"ZC"; TAB( 38)"ZB"
450  GOSUB 650
460  VTAB 16
470  INPUT "ENTER VALUE FOR ZA ";ZA$: CALL  - 998: CALL  - 868
480  IF  VAL (ZA$) = 0 THEN 470
490  INPUT "ENTER VALUE FOR ZB ";ZB$: CALL  - 998: CALL  - 868
500  IF  VAL (ZB$) = 0 THEN 490
510  INPUT "ENTER VALUE FOR ZC ";ZC$: CALL  - 998: CALL  - 868
520  IF  VAL (ZC$) = 0 THEN 510
530  ZA =  VAL (ZA$):ZB =  VAL (ZB$):ZC =  VAL (ZC$)
540  Z = ZA + ZB + ZC
550  Z1 = ZB * ZA / Z
560  Z2 = ZB * ZC / Z
570  Z3 = ZA * ZC / Z
580  PRINT : PRINT : PRINT "THEN...": PRINT
590  PRINT "ZA = ";ZA;: HTAB 20: PRINT "Z1 = ";Z1
600  PRINT "ZB = ";ZB;: HTAB 20: PRINT "Z2 = ";Z2
610  PRINT "ZC = ";ZC;: HTAB 20: PRINT "Z3 = ";Z3
620  PRINT : PRINT "CHANGE VALUES ? ";: GET Y$: HTAB 1: IF Y$ <  > "Y" THEN
     800
630  VTAB 16: CALL  - 868: GOTO 460
640  GOTO 460
650  FOR I = 1 TO 150
660  READ X
670  IF X = 99 THEN 700
680  POKE X,32
690  NEXT I
700  RETURN
710  DATA  1412,1413,1414,1415,1421,1422,1423,1424,1436,1437,1438,1439
720  DATA  1538,1539,1540,1543,1544,1545,1546,1547,1548,1549,1552,1553,15
     54,1559,1560,1561,1562,1563,1564,1567,1568,1569,1570,1571,1572
730  DATA  1668,1669,1670,1671,1674,1677,1678,1679,1680,1689,1692,1693,16
     94,1695,1698
740  DATA  1802,1817,1826,1929,1930,1931,1944,1945,1946,1953,1954,1955,10
     73,1075,1088,1090,1097,1099
750  DATA  1201,1203,1216,1218,1225,1227,1329,1330,1331,1344,1345,1346,13
     53,1354,1355
760  DATA  1458,1473,1482,1586,1601,1610,1706,1707,1708,1709,1710,1711,17
     12,1713,1714,1715,1716,1717,1718,1719,1720,1721,1722
770  INPUT "ENTER VALUE FOR Z3 ";Z3$: CALL  - 998: CALL  - 868
780  DATA  1727,1728,1729,1730,1731,1732,1733,1734,1735,1736,1737,1738,17
     39,1740
790  DATA  99
800  HOME : VTAB 12: PRINT "GOODBYE ....."
```

Fig. 5-21 (cont). Listing for TEEPI program.

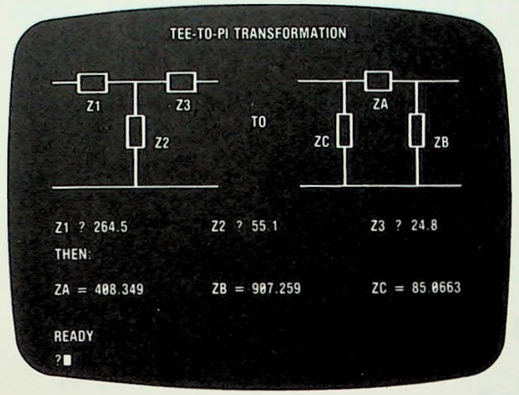


Fig. 5-22. Video results of Tee-to-Pi transformation.

Example 5-11

Determine the mesh currents I_a, I_b, and I_c for the network shown in Fig. 5-24, whose three mesh equations can be shown to be written as:

$$(30)I_a - (10)I_b - (20)I_c = 10$$
$$-(10)I_a + (90)I_b - (50)I_c = 0$$
$$-(20)I_a - (50)I_b + (110)I_c = 0$$

Fig. 5-25 shows the computed solutions for these equations so that I_a, I_b, and I_c are 0.477419, 0.135484, and 0.148387 amp respectively. It is then an easy manner to apply Ohm's law to determine the current through or the voltage across each resistor. For example, the current through the 20-ohm resistor is 0.477419 − 0.148387, or 0.329032 amp, while the voltage drop across it is (0.329032) (20), or 6.58 volts.

```
100  REM  SOLUTION OF SIMULTANEOUS EQUATIONS (SIMULEQU)
101  HOME : VTAB (7): PRINT  TAB( 14): INVERSE : PRINT "SOLUTION OF": NORMAL
102  PRINT : PRINT  TAB( 5): INVERSE : PRINT "SIMULTANEOUS LINEAR EQUATIO
     NS": NORMAL
103  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
104  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
105  FOR I = 1 TO 2500: NEXT I: CLEAR : DIM A(20,21),X(20): HOME
106  INPUT "NUMBER OF EQUATIONS OR VARIABLES? ";N
107  PRINT : INPUT "MINIMUM VALUE FOR ANY SOLUTION? ";MV
108  M = N + 1
109  FOR I = 1 TO N
110  PRINT : PRINT "COEFFICIENTS FOR EQUATION ";I;":"
111  FOR J = 1 TO M
112  PRINT "   #";J;
113  INPUT " ? ";A(I,J)
114  NEXT J
115  NEXT I
116  KK = 0:JJ = 0
117  FOR I = 1 TO N
118  JJ = KK + 1:LL = JJ:KK = KK + 1
119  IF ( ABS (A(JJ,KK)) - MV) <  = 0 THEN 121
120  GOTO 123
121  JJ = JJ + 1
122  GOTO 119
123  IF (LL - JJ) = 0 THEN 129
124  FOR MM = 1 TO M
125  Z = A(LL,MM)
126  A(LL,MM) = A(JJ,MM)
127  A(JJ,MM) = Z
128  NEXT MM
129  D = A(I,I)
130  FOR J = I TO M
131  A(I,J) = A(I,J) / D
132  NEXT J
133  K = I + 1
134  IF (K - M) < 0 THEN 136
```

Continued on next page.

Fig. 5-23. Listing for SIMULEQU program.

```
135  GOTO 143
136  FOR L = K TO N
137  Y = A(L,I)
138  FOR J = I TO M
139  A(L,J) = A(L,J) - A(I,J) * Y
140  NEXT J
141  NEXT L
142  NEXT I
143  X(N) = A(N,M)
144  L = N
145  FOR J = 2 TO N
146  S = 0
147  I = M + 1 - J
148  FOR K = I TO N
149  S = S + A(I - 1,K) * X(K)
150  NEXT K
151  L = L - 1
152  X(L) = A(I - 1,M) - S
153  NEXT J
154  PRINT : PRINT "THE SOLUTIONS ARE:"
155  FOR I = 1 TO N
156  PRINT "   X(";I;") = ";X(I)
157  NEXT I
158  END
```

Fig. 5-23 (cont). Listing for SIMULEQU program.

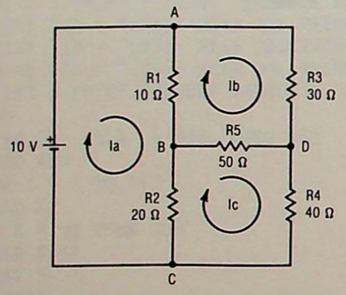
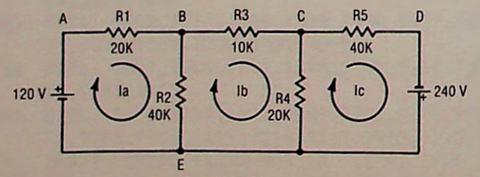


Fig. 5-24. Mesh currents network.



LOOP CURRENTS, Ia, Ib, AND Ic ARE ARBITRARILY TAKEN AS CLOCKWISE.

Fig. 5-26. Circuit for Example 5-12.

```
NUMBER OF EQUATIONS OR VARIABLES? 3

MINIMUM VALUE FOR ANY SOLUTION? 1E-6

COEFFICIENTS FOR EQUATION 1:
   #1 ? 30
   #2 ? -10
   #3 ? -20
   #4 ? 10

COEFFICIENTS FOR EQUATION 2:
   #1 ? -10
   #2 ? 90
   #3 ? -50
   #4 ? 0

COEFFICIENTS FOR EQUATION 3:
   #1 ? -20
   #2 ? -50
   #3 ? 110
   #4 ? 0


THE SOLUTIONS ARE:
   X(1) = .477419355
   X(2) = .135483871
   X(3) = .148387097
```

Fig. 5-25. Computed solutions for mesh currents.


```
NUMBER OF EQUATIONS OR VARIABLES? 5

MINIMUM VALUE FOR ANY SOLUTION? 1E-6

COEFFICIENTS FOR EQUATION 1:
   #1 ? 1
   #2 ? -1
   #3 ? 0
   #4 ? 0
   #5 ? 0
   #6 ? 0

COEFFICIENTS FOR EQUATION 2:
   #1 ? 0
   #2 ? 0
   #3 ? 1
   #4 ? -1
   #5 ? -1
   #6 ? 0

COEFFICIENTS FOR EQUATION 3:
   #1 ? 0
   #2 ? 0
   #3 ? 0
   #4 ? -20000
   #5 ? 40000
   #6 ? 240

COEFFICIENTS FOR EQUATION 4:
   #1 ? 0
   #2 ? -40000
   #3 ? 0
   #4 ? 20000
   #5 ? 0
   #6 ? 0

COEFFICIENTS FOR EQUATION 5:
   #1 ? 20000
   #2 ? 40000
   #3 ? 0
   #4 ? 0
   #5 ? 0
   #6 ? 120


THE SOLUTIONS ARE:
   X(1) = 4.90909091E-03
   X(2) = 5.45454547E-04
   X(3) = 4.36363637E-03
   X(4) = -1.09090909E-03
   X(5) = 5.45454545E-03
```

Fig. 5-27. Computed solutions for branch currents.

Example 5-12

Using the network of Fig. 5-26, determine the 5 branch currents using Kirchhoff's current and voltage laws.

At nodes A and B, application of the current law gives the two equations:

$$I_1 - I_2 - I_3 = 0$$
$$I_3 - I_4 - I_5 = 0$$

while the voltage law yields the following three equations:

$$-(20000)I_4 + (40000)I_5 = 240$$
$$-(40000)I_2 + (10000)I_3 + (20000)I_4 = 0$$
$$(20000)I_1 + (40000)I_2 = 120$$

so that there are 5 equations with 5 unknown currents. Fig. 5-27 shows the 5 computed solutions for this network.

The SIMULEQU program is not valid for complex coefficients, which result from circuits having capacitors and/or inductors in addition to resistors and voltage sources. However, simultaneous equations with complex coefficients are solved using the CPLXSEQ program listed in Fig. 5-28.

When run, the program requires the number of equations which is the same number of unknown variables and the minimum allowable value for any solution, which usually is set at 1E−6 (10^{-6}). The real and imaginary parts of each coefficient are then entered, one at a time for each equation, with the constant term entered last, as shown in the following example.

Example 5-13

For the circuit shown in Fig. 5-29, determine the three loop currents. Writing Kirchhoff's voltage law in rectangular form, we have:

$$(7+3j)I_1 - (5j)I_2 - (5)I_3 = 10/0 = 10$$
$$-(5j)I_1 + (12+3j)I_2 - (2-2j)I_3 = -5/30 = 4.33 - 2.5j$$
$$-(5)I_1 - (2-2j)I_2 + (17-2j)I_3 = -10/90 = -10j$$

Fig. 5-30 shows the computed results that were obtained.

```
100  REM  SIMULTANEOUS EQUATIONS WITH COMPLEX COEFFICIENTS (CPLXSEQ)
101  HOME : VTAB (7): PRINT  TAB( 9): INVERSE : PRINT "SIMUTANEOUS EQUATI
     ONS": NORMAL
102  PRINT : PRINT  TAB( 7): INVERSE : PRINT "WITH COMPLEX COEFFICIENTS":
     NORMAL
103  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
104  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
105  FOR I = 1 TO 2500: NEXT I
106  DIM A(10,22),B(20,22),P(10)
107  HOME : INPUT "NUMBER OF EQUATIONS? ";N
108  PRINT : INPUT "MINIMUM VALUE FOR ANY SOLUTION ? ";EP
109  MA = 2 * N - 1:MB = 2 * N + 1:MC = 2 * N:MD = 2 * N + 2
110  PRINT : PRINT "ENTER:": PRINT
111  FOR I = 1 TO N
112  PRINT "FOR EQUATION #";I
113  FOR J = 1 TO MD STEP 2
114  IF J = MD - 1 THEN 117
115  PRINT : PRINT "   COEFFICIENT "; INT (J / 2) + 1;": REAL PART ";
116  GOTO 118
117  PRINT : PRINT "   CONSTANT TERM: REAL PART ";
118  INPUT " ? ";A(I,J): PRINT  TAB( 19)"IMAG. PART ";
119  INPUT "? ";A(I,J + 1)
120  NEXT J
121  NEXT I
122  IK =  - 1
123  HOME : VTAB (9): PRINT  TAB( 8): FLASH : PRINT "COMPUTATIONS IN PROG
     RESS": NORMAL
124  FOR I = 1 TO MA STEP 2
125  IK = IK + 1
126  FOR J = 1 TO MB STEP 2
127  LJ = I - IK
128  B(I,J) = A(LJ,J):B(I,J + 1) =  - A(LJ,J + 1): NEXT J
129  NEXT I
130  FOR I = 2 TO MC STEP 2
131  FOR J = 1 TO MB STEP 2
132  LJ = I / 2:B(I,J) = A(LJ,J + 1):B(I,J + 1) = A(LJ,J): NEXT J
133  NEXT I
134  JJ = 1
135  FOR I = 1 TO MC
136  KK = I - 1
137  IF  ABS (B(I,I)) - EP <  > 0 THEN 144
138  KK = KK + 1:LL = 0
139  LL = LL + 1
140  FOR J = 1 TO MB
141  BT = B(KK,J):KL = KK + LL:B(KK,J) = B(KL,J)
142  B(KL,J) = BT: NEXT J
143  IF  ABS (B(I,I)) - EP = 0 THEN 139
144  BT = B(I,I)
145  FOR K = 1 TO MB:B(I,K) = B(I,K) / BT: NEXT K
146  JJ = JJ + 1
147  FOR J = JJ TO MC:BT = B(J,I)
148  FOR K = 1 TO MB:B(J,K) = B(J,K) - BT * B(I,K): NEXT K
149  NEXT J
150  NEXT I
151  B(MC,MB) = B(MC,MB) / B(MC,MC):B(MC,MC) = 1:JJ = 1:KK = 2
152  FOR I = 1 TO JJ
```

Continued on next page.

Fig. 5-28. Listing for CPLXSEQ program.


```
153  BT = B(I,KK)
154  FOR J = KK TO MB
155  B(I,J) = B(I,J) - BT * B(KK,J): NEXT J
156  NEXT I
157  JJ = JJ + 1:KK = KK + 1
158  IF JJ - MC <  > 0 THEN 152
159  K = 0
160  HOME : PRINT : PRINT "THE SOLUTIONS ARE:": PRINT
161  FOR I = 1 TO MC STEP 2
162  K = K + 1
163  PRINT "VALUE X(";K;"):"
164  PRINT : PRINT "REAL"; TAB( 10)"IMAGINARY"; TAB( 23)"MAGNITUDE"; TAB(
     34)"PHASE"
165  X(K) =  SQR ((B(I + 1,MB) ^ 2) + (B(I,MB) ^ 2))
166  B(I,MB) =  INT (B(I,MB) * 1E5 + .5) / 1E5:B(I + 1,MB) =  INT (B(I + 1
     ,MB) * 1E5 + .5) / 1E5:X(K) =  INT (X(K) * 1E5 + .5) / 1E5
167  PA =  ATN (B(I + 1,MB) / B(I,MB)) * 57.92577
168  IF B(I,MB) < 0 THEN 170
169  GOTO 171
170  P(K) = PA + 180: GOTO 172
171  P(K) = PA
172  P(K) =  INT (P(K) * 100 + .5) / 100
173  PRINT B(I,MB); TAB( 11);B(I + 1,MB); TAB( 24);X(K); TAB( 34);P(K)
174  PRINT : NEXT I
175  PRINT : END
```

Fig. 5-28 (cont). Listing for CPLXSEQ program.

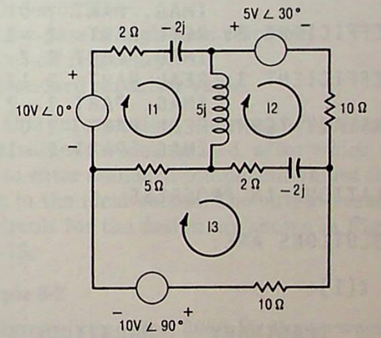


Fig. 5-29. Loop currents network.

```
NUMBER OF EQUATIONS? 3
MINIMUM VALUE FOR ANY SOLUTION? 1E-6
ENTER:
FOR EQUATION #1
   COEFFICIENT 1: REAL PART  ? 7
                  IMAG. PART ? 3
   COEFFICIENT 2: REAL PART  ? 0
                  IMAG. PART ? -5
   COEFFICIENT 3: REAL PART  ? 5
                  IMAG. PART ? 0
   CONSTANT TERM: REAL PART  ? 10
                  IMAG. PART ? 0

FOR EQUATION #2
   COEFFICIENT 1: REAL PART  ? 0
                  IMAG. PART ? -5
   COEFFICIENT 2: REAL PART  ? 12
                  IMAG. PART ? 3
   COEFFICIENT 3: REAL PART  ? -2
                  IMAG. PART ? 2
   CONSTANT TERM: REAL PART  ? -4.33
                  IMAG. PART ? -2.50

FOR EQUATION #3
   COEFFICIENT 1: REAL PART  ? -5
                  IMAG. PART ? 0
   COEFFICIENT 2: REAL PART  ? -2
                  IMAG. PART ? 2
   COEFFICIENT 3: REAL PART  ? 17
                  IMAG. PART ? -2
   CONSTANT TERM: REAL PART  ? 0
                  IMAG. PART ? -10

CALCULATIONS IN PROGRESS

THE SOLUTIONS ARE:

VALUE X(1):

REAL      IMAGINARY     MAGNITUDE    PHASE
.99953    -.22054        1.02357     -12.58

VALUE X(2):

REAL      IMAGINARY     MAGNITUDE    PHASE
-.27472   .12538         .30198      155.2

VALUE X(3):

REAL      IMAGINARY     MAGNITUDE    PHASE
.34296    -.56568        .66153      -59.42
```

Fig. 5-30. Computed results for loop currents.

6

Active Filter Design

This chapter describes programs for the rapid design of various types of active filters. Included are: the state-variable filter (3 op-amps); second- through sixth-order low-pass and high-pass filters having either Bessel (best delay), Butterworth (maximally flat), or Chebyshev (equal ripple) responses; bandpass filters; and notch filters. Each design determines the performance based on user selected standard resistor and capacitor values. For a more detailed discussion about active filters, either the *Design of Active Filters, With Experiments*, or the *Active-Filter Cookbook* (Howard W. Sams & Co., Inc., Indianapolis, IN 46268) are recommended references.

LOW-PASS AND HIGH-PASS FILTERS

The LPHP program listing shown in Fig. 6-1 enables us to design second- through sixth-order low-pass or high-pass filters having either Bessel, Butterworth, or Chebyshev (1-, 2-, or 3-dB) responses. Shown in Figs. 6-2 through 6-11 are the circuits of the low-pass and high-pass filters. Each second-order section is of the voltage-controlled voltage-source design (often referred to as a Sallen-Key filter) with both the frequency determining resistors R and capacitors C made equal. For the third- and fifth-order filters, the first section is a simple first-order passive network followed by a voltage follower. The damping values for each second-order section are controlled by the resistors RA and RB. Because the program restricts the filter's design to having both the frequency determining resistors and capacitors equal for each second-order section, the overall passband gain will be fixed, being set by the values chosen for the RA and RB resistors.

Example 6-1

Design a third-order, 350-Hz high-pass filter having a 1-dB Chebyshev response. When executed, the LPHP program requires the following initial information:

1. Type of filter.
2. Order of filter.
3. Low-pass or high-pass response.
4. Cut-off frequency.
5. Standard capacitor value.

From this information, the remaining ideal component values are determined, after which we are asked to enter standard component values that are closest to the ideal values. The output results and final circuit for the design are shown in Figs. 6-12 and 6-13.

Example 6-2

Design a sixth-order, 3000-Hz Butterworth low-pass filter. As shown in the output results in Fig. 6-14, we only have to determine the frequency determining resistors and capacitors for the first section, since they are the same for the remaining sections in a Butterworth design. The final circuit is shown in Fig. 6-15.

THE STATE-VARIABLE FILTER

The state-variable filter, shown in Fig. 6-16, uses three operational amplifiers, so that there are the following output responses: second-order low-pass, second-order high-pass, and a single-pole bandpass. Resistors R and capacitors C determine the cutoff (or center) frequency, while resistors RA and RB control the damping (or Q). For the

```
100  REM  LOW AND HIGH-PASS FILTER DESIGN (LPHP)
105  HOME : CLEAR : DIM F$(5),VF(120)
110  GOTO 1365
115  FOR V = 1 TO 110
120  READ VF(V): NEXT V
125  HOME : PRINT : PRINT F$(TY);" FILTER DESIGN"
130  PRINT : PRINT "SELECT:"
135  PRINT : PRINT "   LOW-PASS (1)": PRINT : INPUT "   HIGH-PASS (2) ";F
     R
140  PRINT : INPUT "ORDER OF FILTER ";O
145  IF O < 2 OR O > 6 THEN 140
150  PRINT : INPUT "CUTOFF FREQUENCY IN HZ ";FC
155  ON O - 1 GOTO 165,245,375,535,740
160  REM  2ND ORDER DESIGN
165  HOME : PRINT : PRINT "1ST SECTION (2ND ORDER):"
170  PRINT : INPUT "C IN UF ";C
175  K = VF(((TY - 1) * 22) + 1)
180  IF FR = 1 THEN 190
185  F = FC / K: GOTO 195
190  F = FC * K
195  R = 1000 / (6.28 * F * C): PRINT "   THEN R = "; INT (R * 100 + .5) /
     100;" K-OHMS"
200  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
205  IF Z$ = "Y" THEN 170
210  PRINT : INPUT "RA IN K-OHMS ";RA
215  D = VF((TY - 1) * 22 + 2)
220  RB = RA * (2 - D): PRINT : PRINT "   RB = "; INT (RB * 100 + .5) / 10
     0;" K-OHMS"
225  PRINT : INPUT "ANOTHER VALUE FOR RA (Y/N) ";Z$
230  IF Z$ = "Y" THEN 210
235  GOTO 975
240  REM  3RD ORDER DESIGN
245  HOME : PRINT : PRINT "1ST SECTION (1ST ORDER):"
250  PRINT : INPUT "C IN UF ";C1
255  K = VF(((TY - 1) * 22) + 3)
260  IF FR = 1 THEN 270
265  F1 = FC / K: GOTO 275
270  F1 = FC * K
275  R1 = 1000 / (6.28 * F1 * C1): PRINT : PRINT "   THEN R = "; INT (R1 *
     100 + .5) / 100;" K-OHMS"
280  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
285  IF Z$ = "Y" THEN 250
290  HOME : PRINT : PRINT "2ND SECTION (2ND ORDER):"
295  IF TY <  > 2 THEN 305
300  C2 = C1:R2 = R1: GOTO 345
305  PRINT : INPUT "C IN UF ";C2
310  K2 = VF(((TY - 1) * 22) + 5)
315  IF FR = 1 THEN 325
320  F2 = FC / K2: GOTO 330
325  F2 = FC * K2
330  R2 = 1000 / (6.28 * F2 * C2): PRINT : PRINT "   THEN R = "; INT (R2 *
     100 + .5) / 100;" K-OHMS"
335  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
340  IF Z$ = "Y" THEN 305
345  D = VF((TY - 1) * 22 + 6)
350  PRINT : INPUT "RA IN K-OHMS ";R3:R4 = R3 * (2 - D): PRINT : PRINT "
     THEN RB = "; INT (R4 * 100 + .5) / 100;" K-OHMS"
355  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
```

Continued on next page.

Fig. 6-1. Listing for LPHP program.


```
360  IF Z$ = "Y" THEN 350
365  GOTO 1040
370  REM  4TH ORDER DESIGN
375  HOME : PRINT : PRINT "1ST SECTION (2ND ORDER):"
380  K = VF(((TY - 1) * 22) + 7)
385  PRINT : INPUT "C IN UF ";C1
390  IF FR = 1 THEN 400
395  F1 = FC / K: GOTO 405
400  F1 = FC * K
405  R1 = 1000 / (6.28 * F1 * C1): PRINT : PRINT "   THEN R = "; INT (R1 *
     100 + .5) / 100;" K-OHMS"
410  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
415  IF Z$ = "Y" THEN 385
420  D = VF((TY - 1) * 22 + 8)
425  PRINT : INPUT "RA IN K-OHMS ";R3:R4 = R3 * (2 - D)
430  PRINT : PRINT "   THEN RB = "; INT (R4 * 100 + .5) / 100;" K-OHMS"
435  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
440  IF Z$ = "Y" THEN 425
445  HOME : PRINT : PRINT "2ND SECTION (2ND ORDER):"
450  IF TY <  > 2 THEN 460
455  C2 = C1:R2 = R1:F2 = F1: GOTO 500
460  PRINT : INPUT "C IN UF ";C2
465  K2 = VF(((TY - 1) * 22) + 9)
470  IF FR = 1 THEN 480
475  F2 = FC / K2: GOTO 485
480  F2 = FC * K2
485  R2 = 1000 / (6.28 * F2 * C2): PRINT : PRINT "   THEN R = "; INT (R2 *
     100 + .5) / 100;" K-OHMS"
490  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
495  IF Z$ = "Y" THEN 460
500  D2 = VF((TY - 1) * 22 + 10)
505  PRINT : INPUT "RA IN K-OHMS ";R5:R6 = R5 * (2 - D2)
510  PRINT : PRINT "   THEN RB = "; INT (R6 * 100 + .5) / 100;" K-OHMS"
515  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
520  IF Z$ = "Y" THEN 505
525  GOTO 1115
530  REM  5TH ORDER DESIGN
535  HOME : PRINT : PRINT "1ST SECTION (1ST ORDER):"
540  K = VF(((TY - 1) * 22) + 11)
545  PRINT : INPUT "C IN UF ";C1
550  IF FR = 1 THEN 560
555  F1 = FC / K: GOTO 565
560  F1 = FC * K
565  R1 = 1000 / (6.28 * F1 * C1): PRINT : PRINT "   THEN R = "; INT (R1 *
     100 + .5) / 100;" K-OHMS"
570  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
575  IF Z$ = "Y" THEN 545
580  HOME : PRINT : PRINT "2ND SECTION (2ND ORDER) :"
585  IF TY <  > 2 THEN 595
590  R2 = R1:C2 = C1:F2 = F1: GOTO 635
595  PRINT : INPUT "C IN UF ";C2
600  K2 = (((TY - 1) * 22) + 13)
605  IF FR = 1 THEN 615
610  F2 = FC / K2: GOTO 620
615  F2 = FC * K2
620  R2 = 1000 / (6.28 * F2 * C2): PRINT : PRINT "   THEN R = "; INT (R2 *
     100 + .5) / 100;" K-OHMS"
625  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
```

Continued on next page.

Fig. 6-1 (cont). Listing for LPHP program.

```
630  IF Z$ = "Y" THEN 595
635  D2 = VF((TY - 1) * 22 + 14)
640  PRINT : INPUT "RA IN K-OHMS ";R4:R6 = R4 * (2 - D2): PRINT : PRINT "
     THEN RB = "; INT (R6 * 100 + .5) / 100;" K-OHMS"
645  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
650  IF Z$ = "Y" THEN 640
655  HOME : PRINT : PRINT "3RD SECTION (2ND ORDER):"
660  IF TY <  > 2 THEN 670
665  R3 = R1:C3 = C1:F3 = F1: GOTO 710
670  PRINT : INPUT "C IN UF ";C3
675  K3 = (((TY - 1) * 22) + 15)
680  IF FR = 1 THEN 690
685  F3 = FC / K3: GOTO 695
690  F3 = FC * K3
695  R3 = 1000 / (6.28 * F3 * C3): PRINT : PRINT "   THEN R = "; INT (R3 *
     100 + .5) / 100;" K-OHMS"
700  PRINT : INPUT "ANOTHER VALUE FOR C (Y/N) ? ";Z$
705  IF Z$ = "Y" THEN 670
710  D3 = VF((TY - 1) * 22 + 16)
715  PRINT : INPUT "RA IN K-OHMS";R5:R7 = R5 * (2 - D3): PRINT : PRINT "
     THEN RB = "; INT (R7 * 100 + .5) / 100;" K-OHMS"
720  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
725  IF Z$ = "Y" THEN 715
730  GOTO 1195
735  REM  6TH ORDER DESIGN
740  HOME : PRINT : PRINT "1ST SECTION (2ND ORDER):"
745  K = VF(((TY - 1) * 22) + 17)
750  PRINT : INPUT "C IN UF ";C1
755  IF FR = 1 THEN 765
760  F1 = FC / K: GOTO 770
765  F1 = FC * K
770  R1 = 1000 / (6.28 * F1 * C1): PRINT : PRINT "   THEN R = "; INT (R1 *
     100 + .5) / 100;" K-OHMS"
775  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
780  IF Z$ = "Y" THEN 750
785  D = VF((TY - 1) * 22 + 18)
790  PRINT : INPUT "RA IN K-OHMS ";R4:R7 = R4 * (2 - D)
795  PRINT : PRINT "   THEN RB = "; INT (R7 * 100 + .5) / 100;" K-OHMS"
800  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
805  IF Z$ = "Y" THEN 790
810  HOME : PRINT : PRINT "2ND SECTION (2ND ORDER):"
815  IF TY <  > 2 THEN 825
820  R2 = R1:C2 = C1:F2 = F1: GOTO 865
825  PRINT : INPUT "C IN UF ";C2
830  K2 = VF(((TY - 1) * 22) + 19)
835  IF FR = 1 THEN 845
840  F2 = FC / K2: GOTO 850
845  F2 = FC * K2
850  R2 = 1000 / (6.28 * F2 * C2): PRINT : PRINT "   THEN R = "; INT (R2 *
     100 + .5) / 100;" K-OHMS"
855  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
860  IF Z$ = "Y" THEN 825
865  D2 = VF((TY - 1) * 22 + 20)
870  PRINT : INPUT "RA IN K-OHMS ";R5:R8 = R5 * (2 - D2)
875  PRINT : PRINT "   THEN RB = "; INT (R8 * 100 + .5) / 100;" K-OHMS"
880  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
885  IF Z$ = "Y" THEN 870
890  HOME : PRINT : PRINT "3RD SECTION (2ND ORDER):"
```

Continued on next page.

Fig. 6-1 (cont). Listing for LPHP program.


```
895  IF TY <  > 2 THEN 905
900  F3 = F1:R3 = R1:C3 = C1: GOTO 945
905  PRINT : INPUT "C IN UF ";C3
910  K3 = VF(((TY - 1) * 22) + 21)
915  IF FR = 1 THEN 925
920  F3 = FC / K3: GOTO 930
925  F3 = FC * K3
930  R3 = 1000 / (6.28 * F3 * C3): PRINT : PRINT "   THEN R = "; INT (R3 *
     100 + .5) / 100;" K-OHMS"
935  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
940  IF Z$ = "Y" THEN 905
945  D3 = VF((TY - 1) * 22 + 22)
950  PRINT : INPUT "RA IN K-OHMS ";R6:R9 = R6 * (2 - D3)
955  PRINT : PRINT "   THEN RB = "; INT (R9 * 100 + .5) / 100;" K-OHMS"
960  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
965  IF Z$ = "Y" THEN 950
970  GOTO 1275
975  HOME : PRINT "DESIGN SUMMARY:": PRINT "2ND ORDER ";F$(TY);" FILTER":
     PRINT
980  R =  INT (R * 100 + .5) / 100:RA =  INT (RA * 100 + .5) / 100:RB =  INT
     (RB * 100 + .5) / 100
985  PRINT : PRINT  TAB( 15)"1ST": PRINT  TAB( 13)"SECTION": PRINT
990  PRINT "C (UF)"; TAB( 13)C
995  PRINT "R (K-OHMS)"; TAB( 13)R
1000 PRINT "RA (K-OHMS)"; TAB( 13)RA
1005 PRINT "RB (K-OHMS)"; TAB( 13)RB: PRINT
1010 PRINT : PRINT "SELECT STANDARD VALUES FOR:": PRINT
1015 INPUT "RA (IN K-OHMS) ";RA
1020 INPUT "RB (IN K-OHMS) ";RB
1025 G = 1 + (RB / RA):DB = 20 *  LOG (G) /  LOG (10)
1030 PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
     " DB"
1035 GOTO 1495
1040 HOME : PRINT "DESIGN SUMMARY:": PRINT "3RD ORDER ";F$(TY);" FILTER"
     : PRINT
1045 R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100:R3 =
     INT (R3 * 100 + .5) / 100:R4 =  INT (R4 * 100 + .5) / 100
1050 PRINT : PRINT  TAB( 15)"1ST"; TAB( 25)"2ND"
1055 PRINT  TAB( 13)"SECTION"; TAB( 23)"SECTION": PRINT
1060 PRINT "C (UF)"; TAB( 13)C1; TAB( 23)C2
1065 PRINT "R (K-OHMS)"; TAB( 13)R1; TAB( 23)R2
1070 PRINT "RA (K-OHMS)"; TAB( 23)R3
1075 PRINT "RB (K-OHMS)"; TAB( 23)R4: PRINT
1080 PRINT
1085 PRINT : PRINT "SELECT STANDARD VALUES FOR:": PRINT
1090 INPUT "RA (IN K-OHMS) ";R3
1095 INPUT "RB (IN K-OHMS) ";R4
1100 G = 1 + (R4 / R3):DB = 20 *  LOG (G) /  LOG (10)
1105 PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
     " DB"
1110 GOTO 1495
1115 HOME : PRINT "DESIGN SUMMARY:": PRINT "4TH ORDER ";F$(TY);" FILTER:
     PRINT
1120 R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100:R3 =
     INT (R3 * 100 + .5) / 100
1125 R4 =  INT (R4 * 100 + .5) / 100:R5 =  INT (R5 * 100 + .5) / 100:R6 =
     INT (R6 * 100 + .5) / 100
1130 PRINT : PRINT  TAB( 15)"1ST"; TAB( 25)"2ND"
```

Continued on next page.

Fig. 6-1 (cont). Listing for LPHP program.

```
1135  PRINT  TAB( 13)"SECTION"; TAB( 23)"SECTION": PRINT
1140  PRINT "C (UF)"; TAB( 13)C1; TAB( 23)C2
1145  PRINT "R (K-OHMS)"; TAB( 13)R1; TAB( 23)R2
1150  PRINT "RA (K-OHMS)"; TAB( 13)R3; TAB( 23)R4
1155  PRINT "RB (K-OHMS)"; TAB( 13)R5; TAB( 23)R6: PRINT
1160  PRINT : PRINT "SELECT STANDARD VALUES FOR:"
1165  PRINT
1170  INPUT "RA(1)  (IN K-OHMS) ";R3: INPUT "RA(2)  (IN K-OHMS) ";R4
1175  INPUT "RB(1)  (IN K-OHMS) ";R5: INPUT "RB(2)  (IN K-OHMS) ";R6
1180  G = (1 + (R5 / R3)) * (1 + (R6 / R4)):DB = 20 *  LOG (G) /  LOG (10)
1185  PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
      " DB"
1190  GOTO 1495
1195  HOME : PRINT "DESIGN SUMMARY:": PRINT "5TH ORDER ";F$(TY);" FILTER"
      : PRINT
1200  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100:R3 =
      INT (R3 * 100 + .5) / 100:R4 =  INT (R4 * 100 + .5) / 100
1205  R5 =  INT (R5 * 100 + .5) / 100:R6 =  INT (R6 * 100 + .5) / 100:R7 =
      INT (R7 * 100 + .5) / 100
1210  PRINT : PRINT  TAB( 15)"1ST"; TAB( 25)"2ND"; TAB( 35)"3RD"
1215  PRINT  TAB( 13)"SECTION"; TAB( 23)"SECTION"; TAB( 33)"SECTION": PRINT
1220  PRINT "C (UF)"; TAB( 13)C1; TAB( 23)C2; TAB( 33)C3
1225  PRINT "R (K-OHMS)"; TAB( 13)R1; TAB( 23)R2; TAB( 33)R3
1230  PRINT "RA (K-OHMS)"; TAB( 23)R4; TAB( 33)R5
1235  PRINT "RB (K-OHMS)"; TAB( 23)R6; TAB( 33)R7: PRINT
1240  PRINT : PRINT "SELECT STANDARD VALUES FOR:"
1245  PRINT
1250  INPUT "RA(1)  (IN K-OHMS) ";R4: INPUT "RA(2)  (IN K-OHMS) ";R5
1255  INPUT "RB(1)  (IN K-OHMS) ";R6: INPUT "RB(2)  (IN K-OHMS) ";R7
1260  G = (1 + (R6 / R4)) * (1 + (R7 / R5)):DB = 20 *  LOG (G) /  LOG (10)
1265  PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
      " DB"
1270  GOTO 1495
1275  HOME : PRINT "DESIGN SUMMARY:": PRINT "6TH ORDER ";F$(TY);" FILTER"
      : PRINT
1280  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100:R3 =
      INT (R3 * 100 + .5) / 100
1285  R4 =  INT (R4 * 100 + .5) / 100:R5 =  INT (R5 * 100 + .5) / 100:R6 =
      INT (R6 * 100 + .5) / 100
1290  R7 =  INT (R7 * 100 + .5) / 100:R8 =  INT (R8 * 100 + .5) / 100:R9 =
      INT (R9 * 100 + .5) / 100
1295  PRINT : PRINT  TAB( 15)"1ST"; TAB( 25)"2ND"; TAB( 35)"3RD"
1300  PRINT  TAB( 13)"SECTION"; TAB( 23)"SECTION"; TAB( 33)"SECTION": PRINT
1305  PRINT "C (UF)"; TAB( 13)C1; TAB( 23)C2; TAB( 33)C3
1310  PRINT "R (K-OHMS)"; TAB( 13)R1; TAB( 23)R2; TAB( 33)R3
1315  PRINT "RA (K-OHMS)"; TAB( 13)R4; TAB( 23)R5; TAB( 33)R6
1320  PRINT "RB (K-OHMS)"; TAB( 13)R7; TAB( 23)R8; TAB( 33)R9: PRINT
1325  PRINT : PRINT "SELECT STANDARD VALUES FOR:"
1330  PRINT
1335  INPUT "RA(1)  (IN K-OHMS) ";R4: INPUT "RA(2)  (IN K-OHMS) ";R5: INPUT
      "RA(3)  (IN K-OHMS) ";R6
1340  INPUT "RB(1)  (IN K-OHMS) ";R7: INPUT "RB(2)  (IN K-OHMS) ";R8: INPUT
      "RB(3)  (IN K-OHMS) ";R9
1345  G = (1 + (R7 / R4)) * (1 + (R8 / R5)) * (1 + (R9 / R6)):DB = 20 *  LOG
      (G) /  LOG (10)
1350  PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
      " DB"
1355  GOTO 1495
1360  REM   MENU SELECTION
```

Continued on next page.

Fig. 6-1 (cont). Listing for LPHP program.


```
1365  HOME : VTAB (7): PRINT  TAB( 2): INVERSE : PRINT "LOW-PASS AND HIGH
      -PASS FILTER DESIGN": NORMAL
1370  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)
      "HOWARD M. BERLIN, LTD."
1375  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOF
      T VERSION"
1380  FOR II = 1 TO 2500: NEXT II: HOME
1385  PRINT : PRINT "SELECT TYPE OF FILTER RESPONSE:"
1390  PRINT : PRINT "   1. BESSEL"
1395  PRINT : PRINT "   2. BUTTERWORTH"
1400  PRINT : PRINT "   3. 1-DB CHEBYSHEV"
1405  PRINT : PRINT "   4. 2-DB CHEBYSHEV"
1410  PRINT : PRINT "   5. 3-DB CHEBYSHEV"
1415  PRINT : INPUT "CHOICE? ";TY
1420  IF TY > 5 THEN 1365
1425  F$(1) = "BESSEL":F$(2) = "BUTTERWORTH":F$(3) = "1-DB CHEBYSHEV"
1430  F$(4) = "2-DB CHEBYSHEV":F$(5) = "3-DB CHEBYSHEV"
1435  GOTO 115
1440  REM  DESIGN PARAMETERS
1445  DATA  1.274,1.732,1.328,0,1.454,1.447,1.436,1.916,1.610,1.241,1.557
      ,0
1450  DATA  1.613,1.775,1.819,1.091,1.609,1.959,1.694,1.636,1.910,0.977
1455  DATA  1,1.414,1,0,1,1,1,1.848,1,0.765,1,0
1460  DATA  1,1.618,1,0.618,1,1.932,1,1.414,1,0.518
1465  DATA  0.863,1.045,0.452,0,0.911,0.496,0.502,1.275,0.943,0.281,0.280
      ,0
1470  DATA  0.634,0.714,0.961,0.181,0.347,1.314,0.733,0.455,0.977,0.125
1475  DATA   0.852,0.895,0.322,0,0.913,0.412,0.466,1.088,0.946,0.234,0.22
      3,0
1480  DATA  0.624,0.578,0.964,0.142,0.312,1.121,0.727,0.363,0.976,0.0989
1485  DATA  0.841,0.767,0.299,0,0.916,0.326,0.443,0.929,0.950,0.179,0.178
      ,0
1490  DATA  0.614,0.468,0.967,0.113,0.298,0.958,0.722,0.289,0.975,0.0782
1495  CALL  - 756: END
```

Fig. 6-1 (cont). Listing for LPHP program.

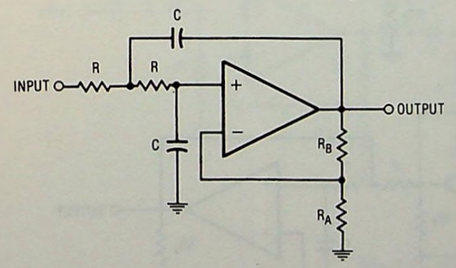


Fig. 6-2. Second-order low-pass filter.

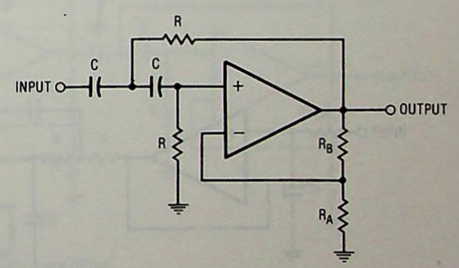


Fig. 6-3. Second-order high-pass filter.

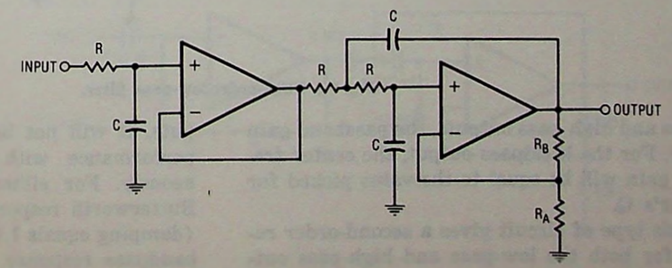


Fig. 6-4. Third-order low-pass filter.

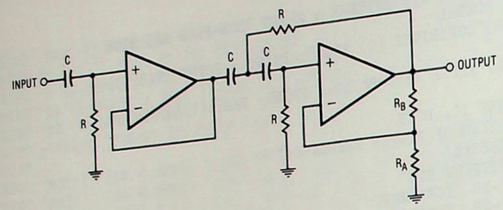


Fig. 6-5. Third-order high-pass filter.

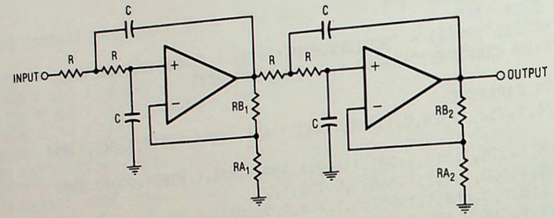


Fig. 6-6. Fourth-order low-pass filter.

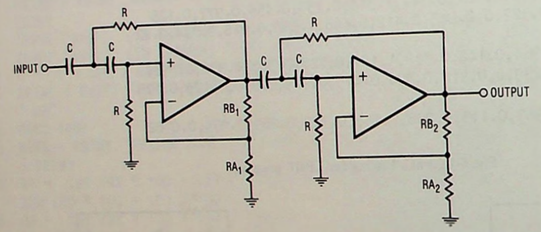


Fig. 6-7. Fourth-order high-pass filter.

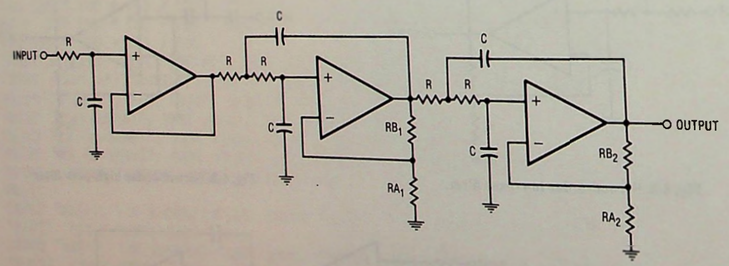


Fig. 6-8. Fifth-order low-pass filter.

low-pass and high-pass outputs, the passband gain is unity. For the bandpass output, the center frequency gain will be equal to the value picked for the filter's Q.

As this type of circuit gives a second-order response for both the low-pass and high-pass outputs, it will not be possible to obtain optimum performance with all three outputs simultaneously. For either a low-pass or a high-pass Butterworth response, Q must be equal to 0.707 (damping equals 1.414, or 1/Q). Consequently the bandpass response suffers terribly! Even for a

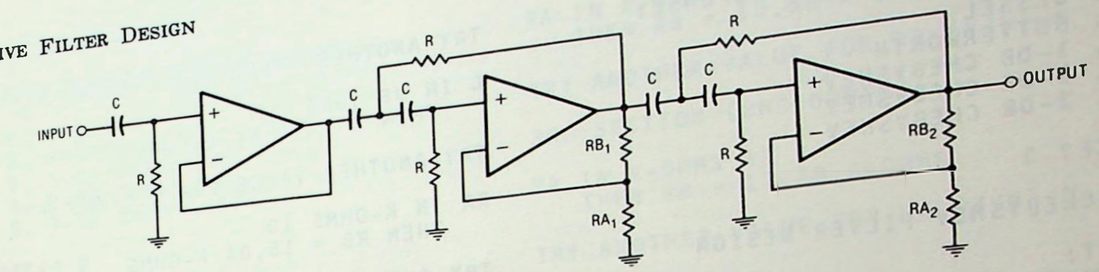


Fig. 6-9. Fifth-order high-pass filter.

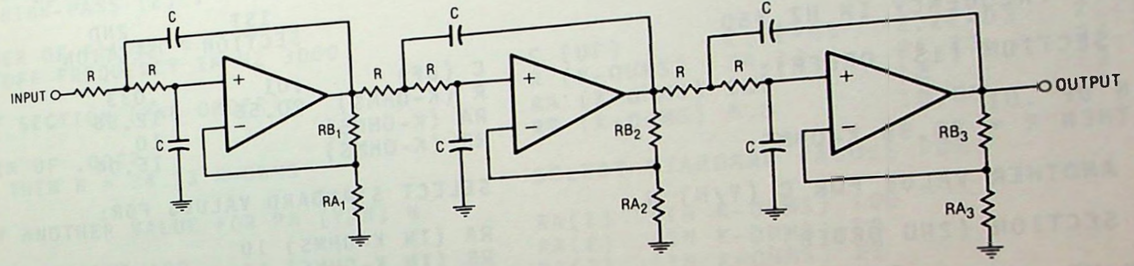


Fig. 6-10. Sixth-order low-pass filter.

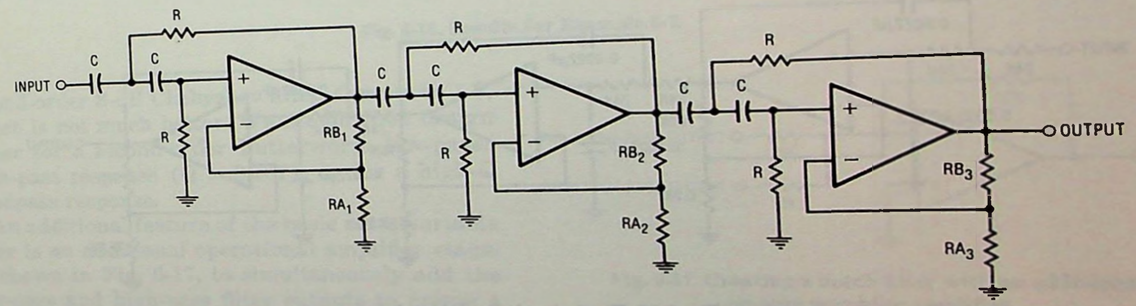


Fig. 6-11. Sixth-order high-pass filter.

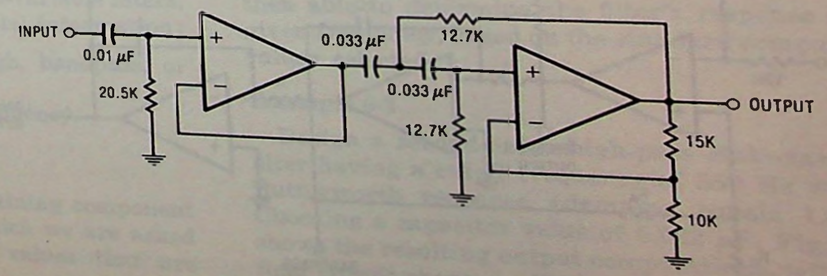


Fig. 6-13. Final circuit for Example 6-1.

```
SELECT TYPE OF FILTER RESPONSE:

   1. BESSEL
   2. BUTTERWORTH
   3. 1-DB CHEBYSHEV
   4. 2-DB CHEBYSHEV
   5. 3-DB CHEBYSHEV

CHOICE? 3

1-DB CHEBYSHEV FILTER DESIGN

SELECT:
   LOW-PASS (1)
   HIGH-PASS (2) 2

ORDER OF FILTER 3
CUTOFF FREQUENCY IN HZ 350

1ST SECTION (1ST ORDER):

C IN UF .01
   THEN R = 20.56 K-OHMS

TRY ANOTHER VALUE FOR C (Y/N) N

2ND SECTION (2ND ORDER):

C IN UF .022
   THEN R= 18.84 K-OHMS

TRY ANOTHER VALUE FOR C (Y/N) Y

C IN UF .033
   THEN R = 12.56 K-OHMS

TRY ANOTHER VALUE FOR C (Y/N) N

RA IN K-OHMS 10
   THEN RB = 15.04 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

DESIGN SUMMARY:
3RD ORDER 1-DB CHEBYSHEV FILTER

                  1ST       2ND
                SECTION   SECTION

C (UF)          .01       .033
R (K-OHMS)      20.56     12.56
RA (K-OHMS)               10
RB (K-OHMS)               15.04

SELECT STANDARD VALUES FOR:

RA (IN K-OHMS) 10
RB (IN K-OHMS) 15

PASSBAND GAIN FIXED AT 7.96 DB
```

Fig. 6-12. Output results for Example 6-1.

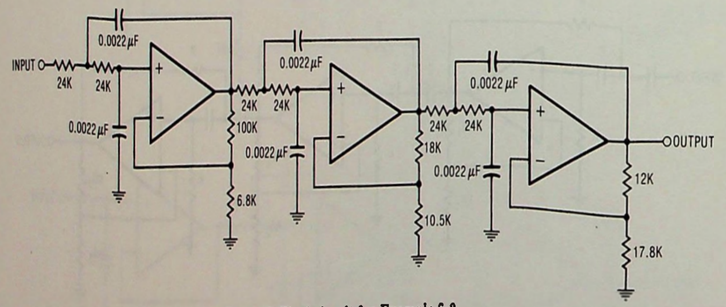


Fig. 6-15. Final circuit for Example 6-2.

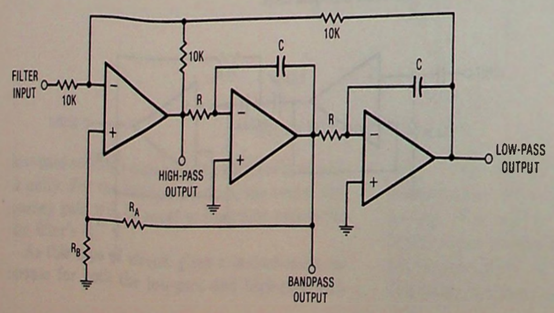


Fig. 6-16. Basic state-variable filter.


```
SELECT TYPE OF FILTER RESPONSE:

   1. BESSEL
   2. BUTTERWORTH
   3. 1-DB CHEBYSHEV
   4. 2-DB CHEBYSHEV
   5. 3-DB CHEBYSHEV

CHOICE? 2

BUTTERWORTH FILTER DESIGN

SELECT:
   LOW-PASS (1)
   HIGH-PASS (2) 1

ORDER OF FILTER 6
CUTOFF FREQUENCY IN HZ 3000

1ST SECTION (2ND ORDER):

C IN UF .0022
   THEN R = 24.13 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

RA IN K-OHMS 100
   THEN RB = 6.8 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

2ND SECTION (2ND ORDER):

TRY ANOTHER VALUE FOR C (Y/N) N

3RD SECTION (2ND ORDER):

RA IN K-OHMS 12
   THEN RB = 17.78 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

DESIGN SUMMARY:
6TH ORDER BUTTERWORTH FILTER

                  1ST       2ND       3RD
                SECTION   SECTION   SECTION

C (UF)          2.2E-03   2.2E-03   2.2E-03
R (K-OHMS)      24.13     24.13     24.13
RA (K-OHMS)     100       18        12
RB (K-OHMS)     6.8       10.55     17.78

SELECT STANDARD VALUES FOR:

RA(1)   (IN K-OHMS) 100
RA(2)   (IN K-OHMS) 18
RA(3)   (IN K-OHMS) 12
RB(1)   (IN K-OHMS) 6.8
RB(2)   (IN K-OHMS) 10.5
RB(3)   (IN K-OHMS) 17.8

PASSBAND GAIN FIXED AT 12.46 DB
```

Fig. 6-14. Results for Example 6-2.

second-order 3-dB Chebyshev filter, Q must be 1.3, which is not much better. We should then design either for a second-order Butterworth low-pass/high-pass response (Q = 0.707), or for a high-Q bandpass response.

An additional feature of the basic state-variable filter is an additional operational amplifier stage, as shown in Fig. 6-17, to simultaneously add the low-pass and high-pass filter outputs to create a notch or band reject filter.

Shown in Fig. 6-18 is the program listing (SVFILTER) for designing state-variable filters, which requires the following initial information:

1. Type of response (low, high, bandpass, or notch).
2. Cutoff, center, or notch frequency.
3. Damping factor, or Q.
4. Standard capacitor value.

From this information, the remaining component values are determined, after which we are asked to enter standard component values that are closest to the ideal values. In addition, we are then able to determine the filter's response at a given frequency, based on the standard component values selected.

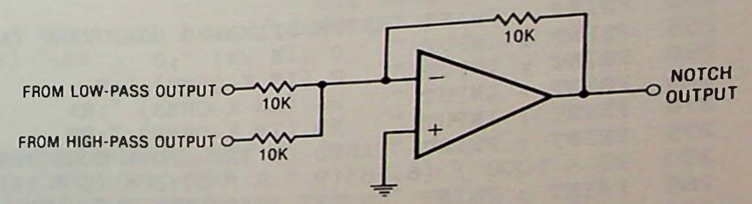


Fig. 6-17. Creating a notch filter with an additional op-amp summing amplifier.

Example 6-3

Design a second-order high-pass state-variable filter having a cutoff frequency of 350 Hz with a Butterworth response (damping equals 1.414). Choosing a capacitor value of 0.022 μF, Fig. 6-19 shows the resulting output computations, with the final circuit shown in Fig. 6-20.

```
100 REM  STATE-VARIABLE FILTER DESIGN (SVFILTER)
105 HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "STATE-VARIABLE FIL
    TER DESIGN": NORMAL
110 PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
    HOWARD M. BERLIN, LTD."
115 PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 12)"APPLESOFT
    VERSION"
120 FOR II = 1 TO 2500: NEXT II: HOME
125 PRINT "OUTPUT RESPONSES AVAILABLE:"
130 PRINT : PRINT "   1. 2ND ORDER LOW PASS."
135 PRINT : PRINT "   2. 2ND ORDER HIGH PASS."
140 PRINT : PRINT "   3. 1-POLE BANDPASS."
145 PRINT : PRINT "   4. 1-POLE NOTCH."
150 PRINT : INPUT "SELECT # ";Z9$
155 IF Z9$ <  > "1" AND Z9$ <  > "2" AND Z9$ <  > "3" AND Z9$ <  > "4" THEN
    150
160 IF Z9$ = "1" THEN 180
165 IF Z9$ = "2" THEN 355
170 IF Z9$ = "3" THEN 535
175 IF Z9$ = "4" THEN 535
180 HOME : PRINT : PRINT "2ND ORDER LOW-PASS SV FILTER"
185 PRINT : INPUT "ENTER CUTOFF FREQUENCY IN HZ ";FC
190 PRINT : INPUT "ENTER DAMPING FACTOR ";D
195 IF D < 3 THEN 205
200 INVERSE : PRINT : PRINT "DAMPING FACTOR TOO LARGE (>3).": PRINT : PRINT
    "PICK ANOTHER VALUE": NORMAL : GOTO 190
205 PRINT : PRINT "PASSBAND GAIN FIXED AT 1.0 (0 DB)"
210 PRINT : INPUT "ENTER CAPACITOR VALUE (IN UF) ";C:P = 6.28319 * FC
215 R = 1000 / (P * C):E = (3 / D) - 1
220 PRINT : PRINT "THEN...": PRINT "R = "; INT (R * 100 + .5) / 100;" K-
    OHMS"
225 PRINT : INPUT "ENTER VALUE FOR RA ";RA
230 RB = RA / E
235 PRINT "THEN...": PRINT "RB = "; INT (RB * 100 + .5) / 100;" K-OHMS"
240 PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";A$
245 IF A$ = "Y" THEN 210
250 PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR:"
255 PRINT : INPUT "   C (IN UF) ";C
260 PRINT : INPUT "   R (IN K-OHMS) ";R
265 PRINT : INPUT "   RA (IN K-OHMS) ";RA
270 PRINT : INPUT "   RB (IN K-OHMS) ";RB
275 PRINT : PRINT "BASED ON THE ABOVE COMPONENTS CHOSEN:"
280 FC = 1000 / (6.28319 * R * C):D = (3 * RB) / (RA + RB)
285 PRINT : PRINT "CUTOFF FREQUENCY = "; INT (FC * 10 + .5) / 10;" HZ"
290 PRINT : PRINT "DAMPING FACTOR = " INT (D * 1000 + .5) / 1000
295 PRINT : PRINT "TRY ANOTHER SET": INPUT "OF STANDARD VALUES (Y/N) ";A
    $
300 IF A$ = "Y" THEN 250
305 HOME : PRINT : INPUT "AMPLITUDE RESPONSE AT A GIVEN FREQUENCY (Y/N)
    ";A$
310 IF A$ = "N" THEN 350
315 PRINT : INPUT "FREQUENCY IN HZ ";FI
320 F = FI / FC
325 DB =  - 20 *  LOG ((F ^ 4) + ((D ^ 2 - 2) * F ^ 2) + 1) /  LOG (10)
330 DB = DB / 2
335 PRINT : PRINT "   AMPLITUDE RESPONSE = "; INT (DB * 100 + .5) / 100;
    " DB"
340 PRINT : INPUT "TRY ANOTHER FREQUENCY (Y/N)? ";A$
```

Continued on next page.

Fig. 6-18. Listing for SVFILTER program.


```
345 IF A$ = "Y" THEN 315
350 PRINT : PRINT "DESIGN COMPLETED": END
355 HOME : PRINT : PRINT "2ND ORDER HIGH-PASS SV FILTER": PRINT
360 PRINT : INPUT "ENTER CUTOFF FREQUENCY IN HZ ";FC
365 PRINT : INPUT "ENTER DAMPING FACTOR ";D
370 IF D < 3 THEN 385
375 INVERSE : PRINT : PRINT "DAMPING FACTOR TOO LARGE (>3).": PRINT "PIC
    K ANOTHER VALUE."
380 NORMAL : GOTO 365
385 PRINT : PRINT "PASSBAND GAIN FIXED AT 1.0 (0 DB)."
390 PRINT : INPUT "ENTER CAPACITOR VALUE IN UF ";C:P = 6.28319 * FC
395 R = 1000 / (P * C):E = (3 / D) - 1
400 PRINT "   THEN R = "; INT (R * 100 + .5) / 100;" K-OHMS"
405 PRINT : INPUT "ENTER VALUE FOR RA (IN K-OHMS) ";RA
410 RB = RA / E
415 PRINT : PRINT "   RB = "; INT (RB * 100 + .5) / 100;" K-OHMS"
420 PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";A$
425 IF A$ = "Y" THEN 390
430 PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR:"
435 PRINT : INPUT "   C (IN UF) ";C
440 PRINT : INPUT "   R (IN K-OHMS) ";R
445 PRINT : INPUT "   RA (IN K-OHMS) ";RA
450 PRINT : INPUT "   RB (IN K-OHMS) ";RB
455 PRINT : PRINT "BASED ON THE ABOVE COMPONENTES CHOSEN:"
460 FC = 1000 / (6.28319 * R * C):D = (3 * RB) / (RA + RB)
465 PRINT : PRINT "CUTOFF FREQUENCY = "; INT (FC * 10 + .5) / 10;" HZ"
470 PRINT : PRINT "DAMPING FACTOR = "; INT (D * 1000 + .5) / 1000
475 PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";A$
480 IF A$ = "Y" THEN 430
485 HOME : PRINT : PRINT "AMPLITUDE RESPONSE": INPUT "AT A GIVEN FREQUEN
    CY (Y/N) ";A$
490 IF A$ = "N" THEN 530
495 PRINT : INPUT "FREQUENCY IN HZ ";FI
500 F = FC / FI
505 DB =  - 20 *  LOG ((F ^ 4) + ((D ^ 2 - 2) * F ^ 2) + 1) /  LOG (10)
510 DB = DB / 2
515 PRINT "   AMPLITUDE RESPONSE = "; INT (DB * 100 + .5) / 100;" DB"
520 PRINT : INPUT "ADDITIONAL FREQUENCIES (Y/N) ";A$
525 IF A$ = "Y" THEN 495
530 PRINT : PRINT "DESIGN COMPLETED": END
535 IF Z9$ = "4" THEN 550
540 HOME : PRINT : PRINT "1-POLE BANDPASS STATE-VARIABLE FILTER"
545 PRINT : INPUT "ENTER CENTER FREQUENCY IN HZ ";FC: GOTO 560
550 HOME : PRINT : PRINT "1-POLE NOTCH STATE-VARIABLE FILTER"
555 PRINT : INPUT "ENTER NOTCH (NULL) FREQUENCY IN HZ ";FC
560 PRINT : INPUT "3 DB BANDWIDTH IN HZ ";BW:Q = FC / BW
565 PRINT : INPUT "ENTER CAPACITOR VALUE IN UF ";C
570 P = 6.28319 * FC
575 R = 1000 / (P * C)
580 E = (3 * Q) - 1
585 PRINT : PRINT "   THEN R = "; INT (R * 100 + .5) / 100;" K-OHMS"
590 PRINT : INPUT "ENTER VALUE FOR RA (IN K-OHMS) ";RA
595 RB = RA / E
600 PRINT : PRINT "   THEN RB = "; INT (RB * 100 + .5) / 100;" K-OHMS"
605 PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";A$
610 IF A$ = "Y" THEN 565
615 PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR:"
```

Continued on next page.

Fig. 6-18 (cont). Listing for SVFILTER program.

```
620  PRINT : INPUT "   C (IN UF) ";C
625  PRINT : INPUT "   R (IN K-OHMS) ";R
630  E = (3 * Q) - 1
635  PRINT : INPUT "   RA (IN K-OHMS) ";RA: PRINT : INPUT "   RB (IN K-OH
     MS) ";RB
640  IF Z9$ = "4" THEN 650
645  PRINT : PRINT "CENTER FREQUENCY GAIN FIXED AT Q ("; INT (Q * 10 + .5
     ) / 10;")": GOTO 655
650  PRINT : PRINT "PASSBAND GAIN FIXED AT 1.0 (0 DB)"
655  FC = 1000 / (6.28319 * R * C):Q = (1 + (RA / RB)) / 3:BW = FC / Q
660  FC =  INT (FC * 100 + .5) / 100:Q =  INT (Q * 10 + .5) / 10:BW =  INT
     (BW * 100 + .5) / 100
665  PRINT : PRINT "BASED ON ABOVE COMPONENT VALUES:"
670  IF Z9$ = "4" THEN 680
675  PRINT : PRINT "CENTER FREQUENCY = ";FC;" HZ": GOTO 685
680  PRINT : PRINT "NOTCH (NULL) FREQUENCY = ";FC;" HZ"
685  PRINT : PRINT "Q = ";Q
690  PRINT : PRINT "3 DB BANDWIDTH = ";BW;" HZ"
695  IF Z9$ = "4" THEN 710
700  DB = 8.68589 *  LOG (Q): PRINT : PRINT "CENTER FREQUENCY GAIN = "; INT
     (DB * 100 + .5) / 100;" DB"
705  GOTO 715
710  PRINT : PRINT "PASSBAND GAIN = 0 DB"
715  PRINT : PRINT "ANOTHER SET": INPUT "OF NOMINAL VALUES (Y/N) ";A$
720  IF A$ = "Y" THEN 615
725  HOME : PRINT : INPUT "AMPLITUDE RESPONSE AT A GIVEN FREQUENCY (Y/N)
     ";A$
730  IF A$ = "N" THEN 815
735  PRINT : INPUT "INPUT FREQUENCY IN HZ ";FI
740  F = FI / FC
745  IF Z9$ = "4" THEN 785
750  DB = 8.6859 *  LOG (Q) - 4.3429 *  LOG (1 + Q ^ 2 * ((F ^ 2 - 1) / F)
     ^ 2)
755  DB =  INT (DB * 100 + .5) / 100
760  PRINT : PRINT "   AMPLITUDE RESPONSE = ";DB;" DB"
765  PRINT : INPUT "ADDITIONAL FREQUENCIES (Y/N)? ";PP$
770  IF PP$ = "Y" THEN 735
775  IF PP$ = "N" THEN 815
780  GOTO 765
785  DB = 4.34294 *  LOG ((1 - F ^ 2) ^ 2) - 4.34294 *  LOG ((1 - F ^ 2) ^
     2 + (F / Q) ^ 2)
790  DB =  INT (DB * 100 + .5) / 100
795  PRINT : PRINT "   AMPLITUDE RESPONSE = ";DB;" DB"
800  PRINT : INPUT "ADDITIONAL FREQUENCIES (Y/N) ? ";PP$
805  IF PP$ = "Y" THEN 735
810  IF PP$ <  > "N" THEN 800
815  PRINT : PRINT "DESIGN COMPLETED"
820  END
```

Fig. 6-18 (cont). Listing for SVFILTER program.

A BANDPASS FILTER FOR Q LESS THAN 10

For values of Q less than 10, the multiple-feedback circuit of Fig. 6-21 is often used, since it uses a minimum of components. Furthermore, the design is simplified by making both frequency determining capacitors equal. Resistors R1 and R3 determine the center frequency passband gain, while the center frequency is set by all three resistors. The program listing (BNDPASS1) is given in Fig. 6-22.

When executed the program requires the following initial information:

1. Design determined by knowing either Q, 3-dB bandwidth, or the 3-dB frequencies.
2. Standard capacitor value.


```
2ND ORDER HIGH-PASS SV FILTER

ENTER CUTOFF FREQUENCY IN HZ 350
ENTER DAMPING FACTOR 1.414
PASSBAND GAIN FIXED AT 1.0 (0 DB)

ENTER CAPACITOR VALUE IN UF .022
   THEN R = 20.67 K-OHMS

ENTER VALUE FOR RA (IN K-OHMS) 27
   RB = 24.07 K-OHMS

TRY ANOTHER SET OF VALUES (Y/N) N

ENTER STANDARD COMPONENT VALUES FOR:
   C (IN UF) .022
   R (IN K-OHMS) 20
   RA (IN K-OHMS) 27
   RB (IN K-OHMS) 24

BASED ON THE ABOVE COMPONENTS CHOSEN:

CUTOFF FREQUENCY = 361.7 HZ
DAMPING FACTOR = 1.412
TRY ANOTHER SET OF VALUES (Y/N) N

AMPLITUDE RESPONSE
AT A GIVEN FREQUENCY (Y/N) Y

FREQUENCY IN HZ 500
   AMPLITUDE RESPONSE = -1.04 DB

ADDITIONAL FREQUENCIES (Y/N) Y

FREQUENCY IN HZ 100
   AMPLITUDE RESPONSE = -22.36 DB

ADDITIONAL FREQUENCIES (Y/N) N

DESIGN COMPLETED
```

Fig. 6-19. Results for Example 6-3.

3. Center frequency.
4. Center frequency voltage gain.

after which the resistor values are determined. In addition, we are then able to determine the filter's response at any frequency based on the standard values chosen. The following example illustrates the BNDPASS1 program.

Example 6-4

Design a bandpass filter with a center frequency of 750 Hz, a voltage gain of 1.3 (+2.3 dB), and a Q of 4.2. Using a standard capacitor value of 0.01 μF, the three resistor values are calculated after which we are asked whether or not we want to try a different set of values, particularly that of the capacitor. If not, we are asked to enter standard component values that are closest to those calculated by the program. Again, we are asked whether or not we want to change the values. If not, then we are asked if we want to know the filter's frequency response at a particular frequency, based on the component values we have just chosen.

Figs. 6-23 and 6-24 show both the output and the final circuit of the program.

Fig. 6-20. Final circuit for Example 6-3.

A BANDPASS FILTER FOR Q LESS THAN 50

The simple multiple-feedback circuit of Fig. 6-21 is not stable for values of Q greater than 10. However, relatively simple bandpass filters having Q values between 10 and 50 are possible with the addition of a second op-amp and three resistors, as shown in Fig. 6-25. Using the BNDPASS2 program of Fig. 6-26, the design of the filter is simplified by making both capacitors equal, as before. In addition, because of the nature of the design of the filter, the center frequency voltage gain must be greater than:

$$G > (Q)^{1/2} \qquad \text{(Eq. 6-1)}$$

so that high gain accompanies high Q.

Example 6-5

Design a bandpass filter having a 3-dB bandwidth of 50 Hz at a center frequency of 1000 Hz (so that Q = 20).

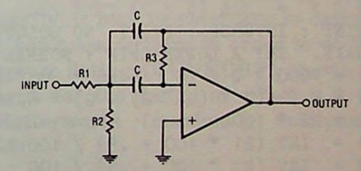
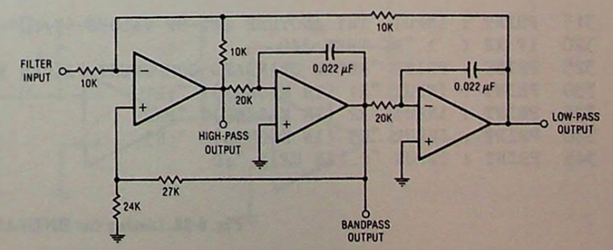


Fig. 6-21. Bandpass filter for Q less than 10.



```
100  REM    ** BANDPASS FILTER DESIGN (BNDPASS1) **
105  CLEAR : CALL  - 936: VTAB (5): PRINT  TAB( 9): INVERSE : PRINT "BAND
     PASS FILTER DESIGN": NORMAL : VTAB (7): PRINT  TAB( 11): INVERSE : PRINT
     "FOR Q LESS THAN 10": NORMAL
110  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980"
115  PRINT : PRINT  TAB( 9)"HOWARD M. BERLIN,LTD."
120  PRINT : PRINT  TAB( 16)"APPLE II"
125  PRINT : PRINT  TAB( 11)"APPLESOFT VERSION"
130  FOR I = 1 TO 2500: NEXT I: CALL  - 936
135  PRINT : PRINT "DESIGN BASED ON KNOWING:": PRINT : PRINT "     1. Q":
     PRINT : PRINT "     2. 3-DB BANDWIDTH": PRINT : PRINT "     3. UPPE
     R & LOWER 3-DB FREQUENCIES"
140  PRINT : INPUT "ENTER CHOICE ?";Z8: IF Z8 > 3 THEN 140
145  ON Z8 GOTO 150,170,180
150  PRINT : INPUT "Q ? ";Q
155  IF Q < 10 THEN 165
160  PRINT : INVERSE : PRINT "Q MUST BE LESS THAN 10": NORMAL : GOTO 150
165  GOTO 230
170  PRINT : INPUT "3-DB BANDWIDTH IN HZ? ";BW
175  GOTO 230
180  PRINT : INPUT "LOWER 3-DB FREQUENCY IN HZ? ";FL
185  PRINT : INPUT "UPPER 3-DB FREQUENCY IN HZ? ";FH
190  F0 =  SQR (FH * FL)
195  PRINT : PRINT "    CENTER FREQUENCY = "; INT (F0 * 10 + .5) / 10;" H
     Z"
200  BW = FH - FL:Q = F0 / BW
205  PRINT : PRINT "    3-DB BANDWIDTH = "; INT (BW * 10 + .5) / 10;" HZ"
210  PRINT : PRINT "    Q = "; INT (Q * 10 + .5) / 10
215  IF Q <  = 10 THEN 225
220  PRINT : INVERSE : PRINT "Q MUST BE LESS THAN 10.": PRINT "INCREASE B
     ANDWIDTH": NORMAL : GOTO 180
225  GOTO 230
230  PRINT : PRINT "HIT RETURN TO CONTINUE";
235  CALL  - 756
240  CALL  - 936: PRINT : INPUT "STANDARD CAPACITOR VALUE IN UF? ";C
245  ON Z8 GOTO 250,250,265
250  PRINT : INPUT "CENTER FREQUENCY IN HZ? ";F0
255  IF Z8 = 1 THEN 265
260  Q = F0 / BW
265  PRINT : INPUT "CENTER FREQUENCY VOLTAGE GAIN? ";G
270  IF Q >  =  SQR (G / 2) THEN 280
275  PRINT : INVERSE : PRINT "NO SOLUTION POSSIBLE FOR STATED Q AND": PRINT
     "GAIN.  TRY A LOWER GAIN": NORMAL : GOTO 265
280  R1 = 1000 * Q / (G * 6.28 * F0 * C):R3 = 2000 * Q / (6.28 * F0 * C)
285  R2 = Q / ((2 * (Q ^ 2) - G) * 6.28 * F0 * C)
290  R2 = R2 * 1000
295  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100
300  R3 =  INT (R3 * 100 + .5) / 100
305  PRINT : PRINT "THEN:": PRINT " R1"," R2"," R3"
310  PRINT : PRINT : PRINT R1,R2,R3: PRINT : PRINT "(ALL VALUES IN K-OHMS
     )"
315  PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N)? ";X$
320  IF X$ <  > "N" THEN 240
325  PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR: "
330  PRINT : INPUT "R1 (IN K-OHMS)? ";R1
335  PRINT : INPUT "R2 (IN K-OHMS)? ";R2
340  PRINT : INPUT "R3 (IN K-OHMS)? ";R3
345  PRINT : INPUT "C (IN UF)? ";C
```

Continued on next page.

Fig. 6-22. Listing for BNDPASS1 program.


```
350  PRINT : PRINT : PRINT "BASED ON THE ABOVE STANDARD COMPONENTS": PRINT
     "CHOSEN:"
355  R3 = 1E3 * R3:R2 = 1E3 * R2:R1 = 1E3 * R1:C = 1E - 6 * C:Y = 1 / (R3 *
     C ^ 2)
360  R1 = 1 / R1:R2 = 1 / R2:F0 =  SQR (Y * (R1 + R2)):F0 = F0 / 6.28
365  G = R3 * R1 / 2
370  Q = G * 6.28 * F0 * C / R1
375  PRINT : PRINT "CENTER FREQUENCY = "; INT (F0 * 10 + .5) / 10;" HZ"
380  PRINT : PRINT "CENTER FREQUENCY VOLTAGE GAIN = "; INT (G * 100 + .5)
      / 100
385  PRINT : PRINT "Q = "; INT (Q * 100 + .5) / 100,"BANDWIDTH = "; INT (
     (F0 / Q) * 100 + .5) / 100;" HZ"
390  PRINT : PRINT "TRY ANOTHER SET OF STANDARD": INPUT "VALUES (Y/N)? ";
     X$
395  IF X$ <  > "N" THEN 330
400  HOME : PRINT : PRINT "DO YOU WISH TO KNOW THE AMPLITUDE": INPUT "RES
     PONSE (Y/N)? ";X$
405  IF X$ <  > "Y" THEN 440
410  PRINT : INPUT "ENTER INPUT FREQUENCY IN HZ? ";FI
415  F = FI / F0:FX = (((F ^ 2) - 1) / F) ^ 2:V = 1 + (Q ^ 2) * FX
420  DB = (20 *  LOG (G) /  LOG (10)) - 10 *  LOG (V) /  LOG (10):DB =  INT
     (DB * 100 + .5) / 100
425  PRINT : PRINT "     AMPLITUDE RESPONSE = ";DB;" DB"
430  PRINT : INPUT "TRY ANOTHER FREQUENCY (Y/N)? ";X$
435  IF X$ = "Y" THEN 410
440  PRINT : PRINT "DESIGN COMPLETED"
445  END
```

Fig. 6-22 (cont). Listing for BNDPASS1 program.

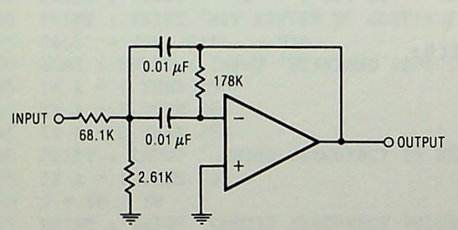
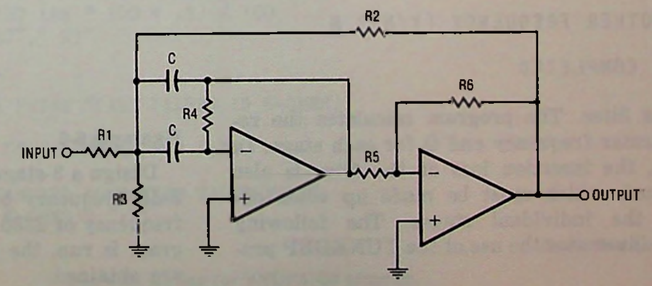


Fig. 6-24. Final circuit for Example 6-4.

Using a standard value of 0.018 μF and a center frequency voltage gain of 5, the output results are shown in Fig. 6-27—the final circuit in Fig. 6-28.

Fig. 6-25. Bandpass filter for Q less than 50.



STAGGERED-TUNED BUTTERWORTH BANDPASS FILTERS

Very often it is required to have a bandpass filter that passes a range of frequencies, instead of a single frequency. In most cases a single stage bandpass section like those previously described will not offer the necessary response. Consequently, it may be necessary to cascade two or more bandpass sections in order to do the job. The program listing (TUNEDBP) shown in Fig. 6-29 allows for the design of either a 2, 3, or 4 stage Butterworth (maximally flat passband)

```
DESIGN BASED ON KNOWING:
     1. Q
     2. 3-DB BANDWIDTH
     3. UPPER & LOWER 3-DB FREQUENCIES

ENTER CHOICE ?1

Q ? 4.2

HIT RETURN TO CONTINUE
STANDARD CAPACITOR VALUE IN UF? .01
CENTER FREQUENCY IN HZ? 750
CENTER FREQUENCY VOLTAGE GAIN? 1.3

THEN:
 R1              R2              R3

68.59           2.62            178.34

(ALL VALUES IN K-OHMS)

TRY ANOTHER SET OF VALUES (Y/N)? N
ENTER STANDARD COMPONENT VALUES FOR:

R1 (IN K-OHMS)? 68.1
R2 (IN K-OHMS)? 2.61
R3 (IN K-OHMS)? 178
C (IN UF)? .01


BASED ON THE ABOVE STANDARD COMPONENTS CHOSEN:
CENTER FREQUENCY = 752.8 HZ
CENTER FREQUENCY VOLTAGE GAIN = 1.31
Q = 4.21        BANDWIDTH = 178.92 HZ

TRY ANOTHER SET OF STANDARD
VALUES (Y/N)? N

DO YOU WISH TO KNOW THE AMPLITUDE
RESPONSE (Y/N)? Y
ENTER INPUT FREQUENCY IN HZ? 400
     AMPLITUDE RESPONSE = -12.9 DB

TRY ANOTHER FREQUENCY (Y/N)? Y

ENTER INPUT FREQUENCY IN HZ? 1125
     AMPLITUDE RESPONSE = -8.83 DB

TRY ANOTHER FREQUENCY (Y/N)? N

DESIGN COMPLETED
```

Fig. 6-23. Results for Example 6-4.

bandpass filter. The program calculates the required center frequency and Q for each stage. In addition, the insertion loss of the filter is also determined, which must be made up when designing the individual stages. The following example illustrates the use of the TUNEDBP program.

Example 6-6

Design a 3-stage bandpass filter having a lower 3-dB frequency of 2050 Hz and an upper 3-dB frequency of 2350 Hz. When the TUNEDBP program is run, the results, as shown in Fig. 6-30, are obtained.


```
100  REM  BANDPASS FILTER DESIGN FOR Q<50 (BNDPASS2)
105  HOME : VTAB (7): PRINT  TAB( 9): INVERSE : PRINT "BANDPASS FILTER DE
     SIGN": NORMAL
110  PRINT : PRINT  TAB( 11): INVERSE : PRINT "FOR Q LESS THAN 50": NORMAL
115  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD. "
120  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
125  FOR I = 1 TO 2500: NEXT I: HOME
130  HOME : PRINT "DESIGN BASED ON KNOWING:": PRINT : PRINT : PRINT  TAB(
     5)"1. Q"
135  PRINT : PRINT  TAB( 5)"2. 3-DB BANDWIDTH"
140  PRINT : PRINT  TAB( 5)"3. UPPER & LOWER 3-DB FREQUENCIES"
145  PRINT : INPUT "ENTER CHOICE? ";A
150  ON A GOTO 155,180,190
155  PRINT : INPUT "Q ? ";Q
160  IF Q >  = 50 THEN 170
165  GOTO 175
170  PRINT : INVERSE : PRINT "Q MUST BE LESS THAN 50.": NORMAL : GOTO 155
175  GOTO 220
180  PRINT : INPUT "3-DB BANDWIDTH IN HZ? ";BW
185  GOTO 220
190  PRINT : INPUT "LOWER 3-DB FREQUENCY IN HZ? ";FL
195  PRINT : INPUT "UPPER 3-DB FREQUENCY IN HZ? ";FH
200  FO =  SQR (FH * FL): PRINT : PRINT "   CENTER FREQUENCY = "; INT (FO *
     10 + .5) / 10;" HZ"
205  BW = FH - FL:Q = FO / BW
210  PRINT : PRINT "   3-DB BANDWIDTH = " INT (BW * 10 + .5) / 10;" HZ"
215  PRINT : PRINT "   Q = "; INT (Q * 100 + .5) / 100
220  PRINT : PRINT "HIT RETURN TO CONTINUE";
225  CALL  - 756: CALL  - 936
230  HOME : PRINT : INPUT "STANDARD CAPACITOR VALUE IN UF? ";C
235  IF A = 1 THEN 250
240  IF A = 3 THEN 265
245  IF A = 2 THEN 250
250  PRINT : INPUT "CENTER FREQUENCY IN HZ? ";FO
255  IF A = 1 THEN 265
260  Q = FO / BW
265  PRINT : INPUT "CENTER FREQUENCY VOLTAGE GAIN? ";G
270  IF G >  SQR (Q) THEN 280
275  PRINT : INVERSE : PRINT "GAIN TOO LOW.  TRY ANOTHER VALUE": NORMAL :
     GOTO 265
280  K = G /  SQR (Q):R1 = Q * 1000 / (6.28 * FO * C)
285  R2 = R1 * K * Q / ((2 * Q) - 1)
290  R6 = K * R1
295  DM = (Q ^ 2) - 1 - (2 / K) + 1 / (K * Q):R3 = R1 / DM
300  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100
305  R3 =  INT (R3 * 100 + .5) / 100:R4 =  INT (R4 * 100 + .5) / 100
310  R5 = R1:R4 = R1:R6 =  INT (R6 * 100 + .5) / 100
315  PRINT : PRINT " R1"," R2"," R3"
320  PRINT R1,R2,R3: PRINT
325  PRINT " R4"," R5"," R6"
330  PRINT R4,R5,R6: PRINT : PRINT "(ALL VALUES IN K-OHMS)"
335  PRINT
340  INPUT "TRY ANOTHER SET OF VALUES (Y/N)? ";X$
345  IF X$ = "Y" THEN 230
350  PRINT : PRINT "ENTER STANDARD COMPONENT VALUE FOR:"
355  PRINT : INPUT " R1 (IN K-OHMS)? ";R1
```

Continued on next page.

Fig. 6-26. Listing for BNDPASS2 program.

```
360  INPUT " R2 (IN K-OHMS)? ";R2
365  INPUT " R3 (IN K-OHMS)? ";R3
370  INPUT " R6 (IN K-OHMS)? ";R6
375  INPUT " C (IN UF)? ";C
380  PRINT : PRINT "BASED ON ABOVE STANDARD COMPONENTS": PRINT "CHOSEN:"
385  K = R6 / R1:XX = (1000 / (6.28 * C * R1)) *  SQR ((R1 + 2 * R3) / R3)
390  FO = XX:Q = 6.28 * R1 * FO * C / 1000
395  G = K *  SQR (Q)
400  PRINT : PRINT "CENTER FREQUENCY = "; INT (FO * 10 + .5) / 10;" HZ"
405  PRINT : PRINT "CENTER FREQUENCY GAIN = "; INT (G * 100 + .5) / 100
410  PRINT : PRINT "Q = "; INT (Q * 100 + .5) / 100
415  PRINT : PRINT "TRY ANOTHER SET OF STANDARD": INPUT "VALUES (Y/N)? ";
     X$
420  IF X$ = "Y" THEN 350
425  HOME : PRINT "DO YOU WISH TO KNOW THE AMPLITUDE": INPUT "RESPONSE (Y
     /N)? ";X$
430  IF X$ = "Y" THEN 440
435  GOTO 470
440  PRINT : INPUT "INPUT FREQUENCY IN HZ? ";FI
445  F = FI / FO:FX = (((F ^ 2) - 1) / F) ^ 2:V = 1 + (Q ^ 2) * FX
450  DB = (20 *  LOG (G) /  LOG (10)) - (10 *  LOG (V) /  LOG (10))
455  PRINT : PRINT "   AMPLITUDE  RESPONSE = "; INT (DB * 100 + .5) / 100
     ;" DB"
460  PRINT : INPUT "TRY ANOTHER FREQUENCY (Y/N)? ";X$
465  IF X$ = "Y" THEN 440
470  PRINT : PRINT "DESIGN COMPLETED"
475  END
```

Fig. 6-26 (cont). Listing for BNDPASS2 program.

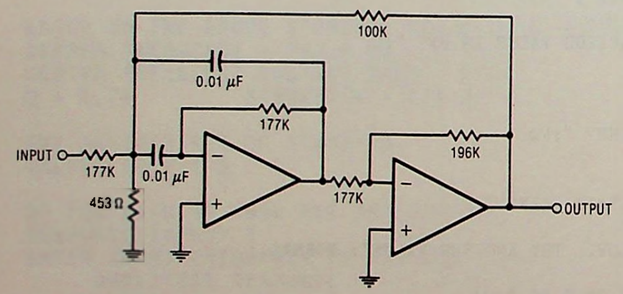


Fig. 6-28. Final circuit for Example 6-5.

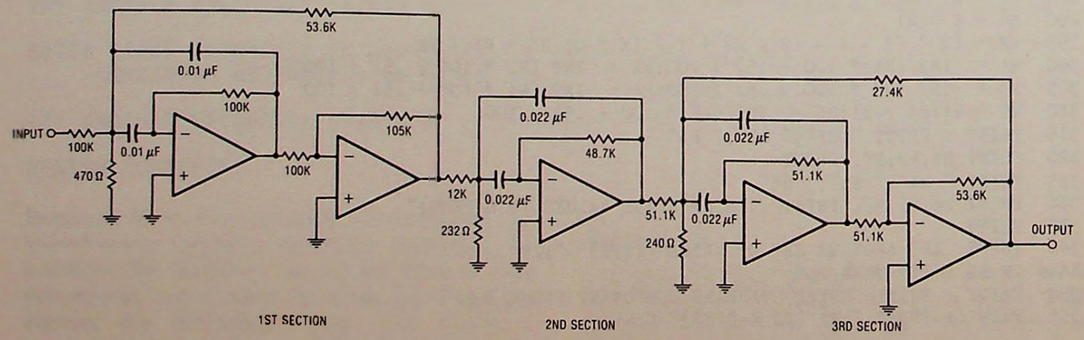


Fig. 6-31. Final circuit for Example 6-6.


```
        BANDPASS FILTER DESIGN FOR Q LESS THAN 50
DESIGN BASED ON KNOWING:
     1. Q
     2. 3-DB BANDWIDTH
     3. UPPER & LOWER 3-DB FREQUENCIES

ENTER CHOICE ?2

3-DB BANDWIDTH IN HZ? 50

HIT RETURN TO CONTINUE
STANDARD CAPACITOR VALUE IN UF? .018
CENTER FREQUENCY IN HZ? 1000

CENTER FREQUENCY VOLTAGE GAIN? 3

GAIN TOO LOW.  TRY ANOTHER VALUE

CENTER FREQUENCY VOLTAGE GAIN? 5

THEN:
 R1              R2              R3
176.93          101.44           .45

 R4              R5              R6
176.93          176.93          197.81

(ALL VALUES IN K-OHMS)

TRY ANOTHER SET OF VALUES (Y/N)? N
ENTER STANDARD COMPONENT VALUE FOR:

 R1 (IN K-OHMS)? 177
 R2 (IN K-OHMS)? 100
 R3 (IN K-OHMS)? 0.453
 R6 (IN K-OHMS)? 196
 C (IN UF)? .018

BASED ON THE ABOVE STANDARD COMPONENTS CHOSEN:
CENTER FREQUENCY = 990.5 HZ
CENTER FREQUENCY GAIN = 4.93
Q = 19.82

TRY ANOTHER SET OF STANDARD
VALUES (Y/N)? N

DO YOU WISH TO KNOW THE AMPLITUDE
RESPONSE (Y/N)? Y

INPUT FREQUENCY IN HZ? 990.5
     AMPLITUDE REPONSE = 13.86 DB

TRY ANOTHER FREQUENCY (Y/N)? Y

INPUT FREQUENCY IN HZ? 300
     AMPLITUDE REPONSE = -21.62 DB
```

Continued on next page.

Fig. 6-27. Results for Example 6-5.

```
360  INPUT " R2 (IN K-OHMS)? ";R2
365  INPUT " R3 (IN K-OHMS)? ";R3
370  INPUT " R6 (IN K-OHMS)? ";R6
375  INPUT " C (IN UF)? ";C
380  PRINT : PRINT "BASED ON ABOVE STANDARD COMPONENTS": PRINT "CHOSEN:"
385  K = R6 / R1:XX = (1000 / (6.28 * C * R1)) *  SQR ((R1 + 2 * R3) / R3)
390  FO = XX:Q = 6.28 * R1 * FO * C / 1000
395  G = K *  SQR (Q)
400  PRINT : PRINT "CENTER FREQUENCY = "; INT (FO * 10 + .5) / 10;" HZ"
405  PRINT : PRINT "CENTER FREQUENCY GAIN = "; INT (G * 100 + .5) / 100
410  PRINT : PRINT "Q = "; INT (Q * 100 + .5) / 100
415  PRINT : PRINT "TRY ANOTHER SET OF STANDARD": INPUT "VALUES (Y/N)? ";
     X$
420  IF X$ = "Y" THEN 350
425  HOME : PRINT "DO YOU WISH TO KNOW THE AMPLITUDE": INPUT "RESPONSE (Y
     /N)? ";X$
430  IF X$ = "Y" THEN 440
435  GOTO 470
440  PRINT : INPUT "INPUT FREQUENCY IN HZ? ";FI
445  F = FI / FO:FX = (((F ^ 2) - 1) / F) ^ 2:V = 1 + (Q ^ 2) * FX
450  DB = (20 *  LOG (G) /  LOG (10)) - (10 *  LOG (V) /  LOG (10))
455  PRINT : PRINT "   AMPLITUDE  RESPONSE = "; INT (DB * 100 + .5) / 100
     ;" DB"
460  PRINT : INPUT "TRY ANOTHER FREQUENCY (Y/N)? ";X$
465  IF X$ = "Y" THEN 440
470  PRINT : PRINT "DESIGN COMPLETED"
475  END
```

Fig. 6-26 (cont). Listing for BNDPASS2 program.

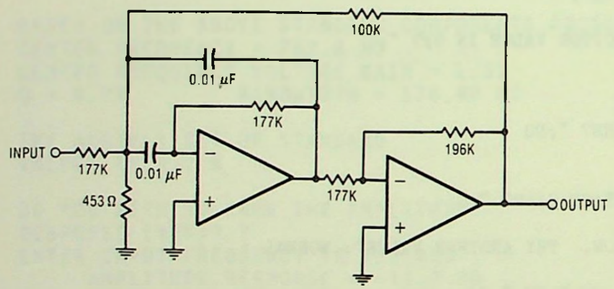


Fig. 6-28. Final circuit for Example 6-5.

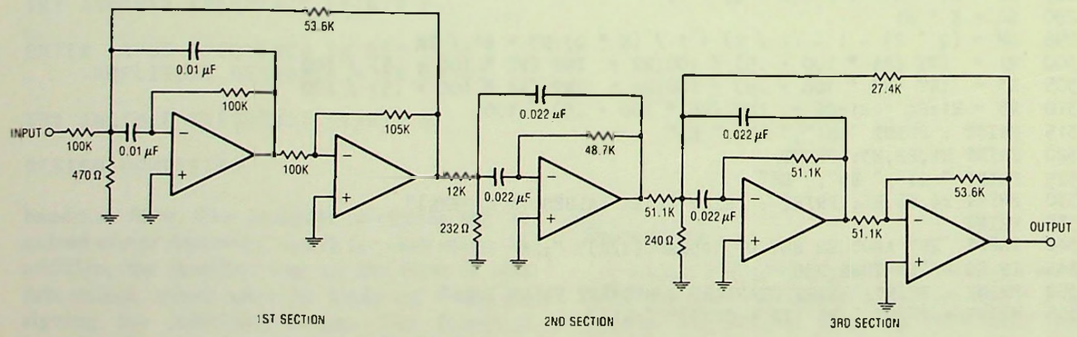


Fig. 6-31. Final circuit for Example 6-6.


```
        BANDPASS FILTER DESIGN FOR Q LESS THAN 50
DESIGN BASED ON KNOWING:
     1. Q
     2. 3-DB BANDWIDTH
     3. UPPER & LOWER 3-DB FREQUENCIES

ENTER CHOICE ?2

3-DB BANDWIDTH IN HZ? 50

HIT RETURN TO CONTINUE
STANDARD CAPACITOR VALUE IN UF? .018
CENTER FREQUENCY IN HZ? 1000

CENTER FREQUENCY VOLTAGE GAIN? 3

GAIN TOO LOW.  TRY ANOTHER VALUE

CENTER FREQUENCY VOLTAGE GAIN? 5

THEN:
 R1              R2              R3
176.93          101.44           .45

 R4              R5              R6
176.93          176.93          197.81

(ALL VALUES IN K-OHMS)

TRY ANOTHER SET OF VALUES (Y/N)? N
ENTER STANDARD COMPONENT VALUE FOR:

 R1 (IN K-OHMS)? 177
 R2 (IN K-OHMS)? 100
 R3 (IN K-OHMS)? 0.453
 R6 (IN K-OHMS)? 196
 C (IN UF)? .018

BASED ON THE ABOVE STANDARD COMPONENTS CHOSEN:
CENTER FREQUENCY = 990.5 HZ
CENTER FREQUENCY GAIN = 4.93
Q = 19.82

TRY ANOTHER SET OF STANDARD
VALUES (Y/N)? N

DO YOU WISH TO KNOW THE AMPLITUDE
RESPONSE (Y/N)? Y

INPUT FREQUENCY IN HZ? 990.5
     AMPLITUDE REPONSE = 13.86 DB

TRY ANOTHER FREQUENCY (Y/N)? Y

INPUT FREQUENCY IN HZ? 300
     AMPLITUDE REPONSE = -21.62 DB
```

Continued on next page.

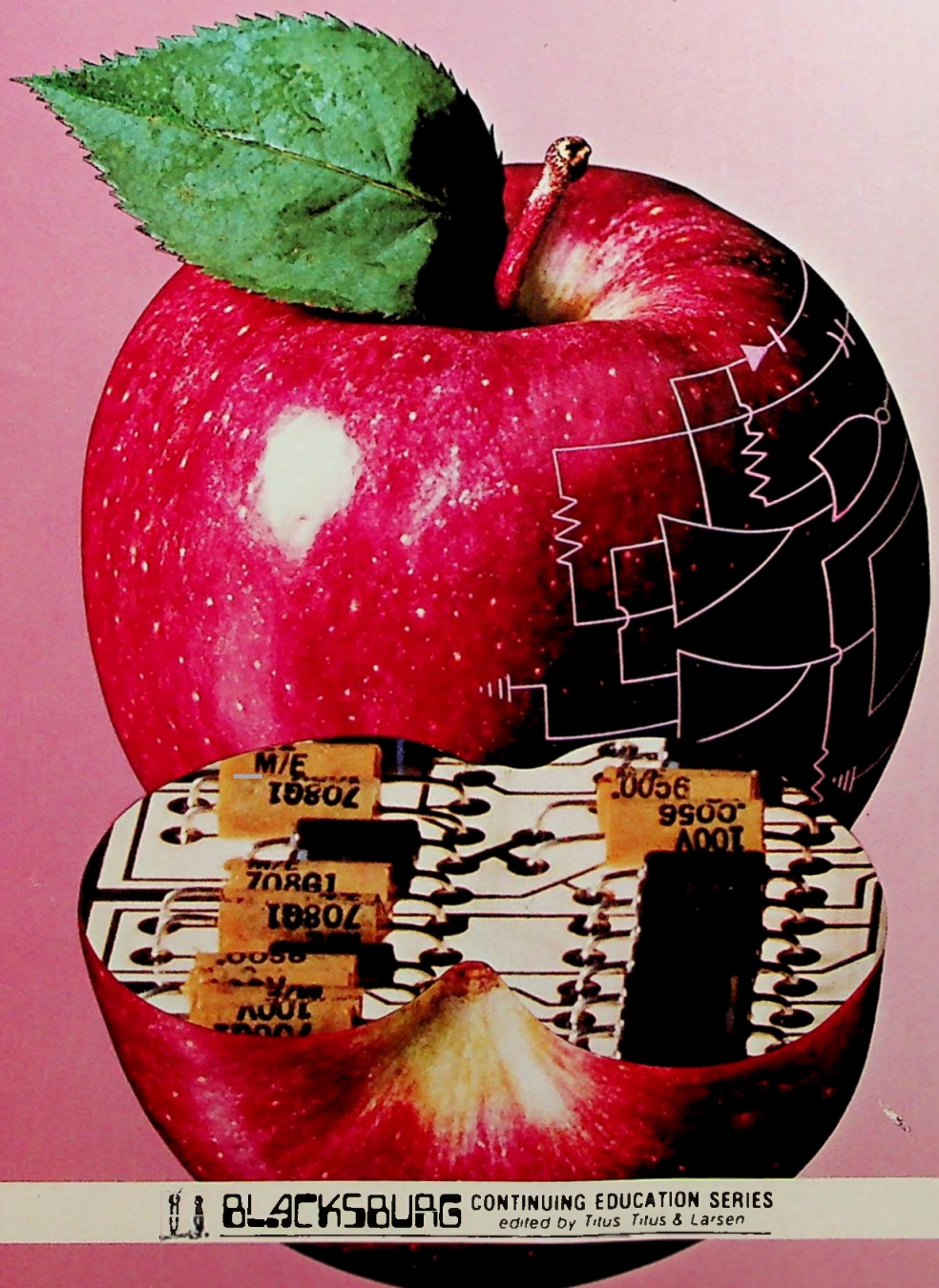
Fig. 6-27. Results for Example 6-5.

21863

CIRCUIT DESIGN PROGRAMS FOR THE APPLE II®

050A 151
PRICE 15.95
2J 2072

BY
HOWARD M. BERLIN



BLACKSBURG CONTINUING EDUCATION SERIES
edited by Titus Titus & Larsen

The Blacksburg Continuing Education™ Series

The Blacksburg Continuing Education Series™ of books provide a Laboratory—or experiment-oriented approach to electronic topics. Present and forthcoming titles in this series include:

- Advanced 6502 Interfacing
- Analog Instrumentation Fundamentals
- Apple Interfacing
- Basic Business Software
- BASIC Programmer's Notebook
- Circuit Design Programs for the Apple II
- Circuit Design Programs for the TRS-80
- Design of Active Filters, With Experiments
- Design of Op-Amp Circuits, With Experiments
- Design of Phase-Locked Loop Circuits, With Experiments
- Design of Transistor Circuits, With Experiments
- Design of VMOS Circuits, With Experiments
- 8080/8085 Software Design (2 Volumes)
- 8085A Cookbook
- Electronic Music Circuits
- 555 Timer Applications Sourcebook, With Experiments
- Guide to CMOS Basics, Circuits, & Experiments
- How to Program and Interface the 6800
- Introduction to FORTH
- Microcomputer—Analog Converter Software and Hardware Interfacing
- Microcomputer Interfacing With the 8255 PPI Chip
- Microcomputer Design and Maintenance
- NCR Basic Electronics Course, With Experiments
- NCR Data Communications Concepts
- NCR Data Processing Concepts Course
- NCR EDP Concepts Course
- PET Interfacing
- Programming and Interfacing the 6502, With Experiments
- 16-Bit Microprocessors
- 6502 Software Design
- 6801, 68701, and 6803 Microcomputer Programming and Interfacing
- The 68000: Principles and Programming
- 6809 Microcomputer Programming & Interfacing, With Experiments
- TEA: An 8080/8085 Co-Resident Editor/Assembler
- TRS-80 Assembly Language Made Simple
- TRS-80 Interfacing (2 Volumes)
- TRS-80 More Than BASIC

In most cases, these books provide both text material and experiments, which permit one to demonstrate and explore the concepts that are covered in the book. These books remain among the very few that provide step-by-step instructions concerning how to learn basic electronic concepts, wire actual circuits, test microcomputer interfaces, and program computers based on popular microprocessor chips. We have found that the books are very useful to the electronic novice who desires to join the "electronics revolution," with minimum time and effort.

Jonathan A. Titus, Christopher A. Titus, and David G. Larsen
"The Blacksburg Group"

Bug symbol trademark Nanotran, Inc., Blacksburg, VA 24060

Circuit Design Programs for the Apple II®

by

Howard M. Berlin

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

Copyright © 1982 by Howard M. Berlin, Ltd.

FIRST EDITION
FIRST PRINTING—1982

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-21863-1
Library of Congress Catalog Card Number: 81-85516

Edited by: *C. Herbert Feltner*
Illustrated by: *T. R. Emrick*

Printed in the United States of America.

Preface

The great mathematician Gottfried Wilhelm Leibniz perhaps stated it best when he said: "It is unworthy of excellent men to lose hours like slaves in the labor of calculation." Before there were computers, virtually every aspect of electronic circuit design and analysis was performed either by hand, or with a slide rule. When done by hand, there was always a good chance that a computational error would exist. With a slide rule, the preciseness of the result left much to be desired.

With the great strides made in the integrated circuit revolution, there now exist a number of relatively inexpensive "personal-type" digital computers, such as the Apple II®, which can efficiently perform virtually any type of computation required for the design and analysis of electronic networks. Consequently, it is the goal of this book to present a variety of useful BASIC language programs which will greatly simplify the design and analysis of commonly encountered circuit problems.

All of the programs listed in this book were written for an Apple II "Plus"® or an Apple II having the Applesoft® firmware card installed. In either case, there should be 32K of random access memory although 48K is preferred. Applesoft BASIC offers many advantages over the integer BASIC set of commands and statements, particularly the mathematical functions, many of which are used repeatedly in the programs described in this book. However, many of these routines can be translated to run on other computer systems such as the PET® and TRS-80®. The program listings and the actual output results, as they would normally appear on the video display, were printed using a "daisywheel" printer instead of one using a dot-matrix font to make the reading easier.

In eight chapters, a variety of programs are presented which enable the user to solve a myriad of problems relating to plotting and the simple statistical verification of experimental data; rms and average values, as well as the Fourier series of a periodic waveform; inverse Laplace, Fourier, and other network transforms; design of matching pads, attenuators, active filters, heat sinks, integrated circuit timer, zener diode regulator, and bipolar transistor circuits; and the solution of simultaneous equations with real or complex coefficients, damped oscillations, and polynomial roots. Some of the programs, such as plot routines using HiRes graphics, are written as subroutines so that they may be used as part of some of your own programs.

All output is displayed on the video screen, but most programs can be easily modified so that numerical results are handled by a line printer. The least squares regression program has the capability to store data on disk with a given file name. In addition, each program is assigned a name, which may be used as the file specification for those of you storing programs on disk media.

Several of the programs in this book were modified from programs originally written in FORTRAN, which were part of extensive subroutine packages. Others were adapted from BASIC programs that have been published elsewhere; however, the algorithms are standard. The mathematical theory for a particular circuit's design, or numerical technique used was intentionally kept to a minimum, since this book was not meant to replace traditional texts on circuit theory or numerical analysis and methods. However, there are short descriptions of the programs, along with computed examples and applicable schematic diagrams, which are used to aid in both the explana-

tion and verification of results against your own copies of the programs.

All the programs listed in this book are available on diskettes, prepared in the 13-sector DOS 3.2 (and earlier) format. In addition, the DOS has been specially modified to allow you to BOOT any one of the seven diskettes on a DOS 3.3 system. Further information on these programs will be found at the end of this book.

HOWARD M. BERLIN

This book is dedicated to Mark Samuel

Contents

CHAPTER 1

ABOUT THE PROGRAMS 7

Basic Facts—Using a Line Printer—Graphics—Program Listings

CHAPTER 2

PLOTTING GRAPHS 9

Histograms—Cartesian Plots—Semilogarithmic Plots—Log-Log Plots—Polar Plots—Plotting with a Line Printer

CHAPTER 3

THE ANALYSIS OF SIGNALS 33

The Average and RMS Values of a Waveform—Fourier Series—Fourier Transform—Analysis of Damped Oscillations

CHAPTER 4

BASIC STATISTICS AND THE ANALYSIS OF DATA 47

Least Squares Regression Curves—Sample Statistics

CHAPTER 5

NETWORKS AND TRANSFORMS 61

Roots of Polynomials—The Inverse Laplace Transform—Impedance Matching Pads—Pi-Tee (Delta-Wye) Transforms—Mesh Current Network Analysis

CHAPTER 6

ACTIVE FILTER DESIGN 83

Low-Pass and High-Pass Filters—The State-Variable Filter—A Bandpass Filter for Q Less Than 50—Staggered-Tuned Butterworth Bandpass Filters—The Notch Filter—A Wideband Filter

CHAPTER 7

SOLID-STATE DEVICES 111

Zener Diode Voltage Regulation Design—555 Timer Design—Transistors—Heat Sink Design

CHAPTER 8

ADDITIONAL ROUTINES 123

4-Quadrant Arctangent Function—Complex Number Math—Minimum and Maximum Values of an Array

APPENDIX

STANDARD RESISTOR AND CAPACITOR VALUES 129

Resistors—Capacitors

INDEX 131

1

About the Programs

BASIC FACTS

The programs described in this book were written for either an Apple II "Plus"® system or an Apple II with the Applesoft® card installed. In either case, the system should have a minimum of 32K of read/write memory, although 48K is preferred. The reasons for this system configuration are as follows:

1. Diskette Applesoft occupies approximately 10K of memory; system overhead and video occupies 2K; while DOS (the disk operating system) occupies another 10.5K. Consequently, diskette Applesoft cannot be used in Apples with less than 24K of memory.
2. Cassette Applesoft has the same memory requirement as diskette Applesoft, except that the DOS requirement is not applicable. Thus, cassette Applesoft cannot be used in Apples with less than 16K of memory.
3. However, the HGR command, used by all the video plotting programs in this book, is not supported by either the cassette or diskette Applesoft.

Although the programs can be stored on cassette tape media, diskettes are preferred for quicker loading of the programs. The REGRESSN program of Chapter 4 was written so that the data may be saved on disk after choosing an appropriate file name.

All computations are performed using single precision arithmetic, so that most numerical results will be displayed with a maximum of nine digits. For the majority of the computations, this has been shown to be more than sufficient. In other cases, the numerical calculations are provided with enough decimal places to make the results significant. As an example, resistance values calculated in kilohms are usually displayed with two decimal places.

USING A LINE PRINTER

Just about all of the programs can be easily converted to have the output results sent to a line printer rather than the video display. Other programs, such as LINEPLOT, XLOGPLOT, and POLARPLT, are specifically designed to plot the results on a line printer. Although primarily intended to be used with parallel type printers, these programs are also easily adapted for serial printers.

GRAPHICS

All video display plotting uses high resolution, or HiRes graphics, while those programs that draw a schematic diagram use LoRes graphics. It should be remembered that page 1 of the HiRes display area begins at decimal memory address 8192. If your program, after the inclusion of any one of the graphics subroutines, becomes too large, you may overlap the HiRes display area in memory and the program may "hang." Before running such a program, type:

```
PRINT PEEK (105) + PEEK (106)*256
```

and then hit the RETURN key. The number displayed is the current LOMEM. If it is greater than 8192, you must then either delete some of the program, add a line at the beginning of your program to set LOMEM to 16384 causing the variables to be stored above the HiRes display area, or you may want to set up a chain to this program so that you use less memory. Pages 141-142 of the DOS 3.2 manual details memory allocation.

PROGRAM LISTINGS

The listings for the programs discussed in this book were printed with a "daisywheel" printer to make the reading easier when compared to a dot-matrix–style printer. There are a number of programs which require exponentiation, which will be represented either by the symbols ^ or © in the listings.

2

Plotting Graphs

Perhaps one of the best methods for describing mathematical functions or experimental results is to plot them in the form of a graph. Remember, a good picture is worth 10,000 words. This chapter illustrates how it is possible to graph functions or discrete data points on the video display in the form of histograms, Cartesian, semilogarithmic, log-log, or polar plots, all of which use HiRes graphics. Most of these programs are primarily written as subroutines so that they may be easily used in conjunction with your own special programs. In addition, three programs are included which allow plotting of various types of graphs on either a serial or parallel line printer.

HISTOGRAMS

One of the simplest types of graphs is the histogram, or bar graph, which is usually used to display frequency distributions. Fig. 2-1 shows the listing of the PHISTGM subroutine used to generate a histogram on the video display. Required input variables are: the number of data points N, the horizontal and vertical values X(I) and Y(I), respectively, for each point, and the graph title T$. Fig. 2-2 illustrates a simple driver, or main program that can be used with the PHISTGM subroutine by executing a GOSUB 5000 statement as in line 155. The subroutine assumes that each X-axis value has only one corresponding Y-axis value. Consequently, a maximum of 265 individual bars may be plotted. The DIM statement in line 105 of the main program must be included because the coordinate values are stored before they are plotted on the display.

Example 2-1

Plot a histogram of the result of measuring the values of a sample of 55, 10% resistors, all of which are supposed to have the same nominal value, 100 ohms. As a result of our sampling, we find that the values vary with several resistors having like values, as listed in Table 2-1.

Using the above data, Fig. 2-3A shows the results of executing the main program of Fig. 2-2 and the PHISTGM subroutine (Fig. 2-1) as one program. After all the 14 individual points and the graph title are entered, the display informs us as to the current ranges of the two axes. The program then asks us how we want the axes scaled, such as having the horizontal axis range from 80 to 120 while the vertical axis varies from 0 to 15. Because of the method in which the axes are scaled, it is best to have the vertical axis (i.e., the "frequency" axis) incremented by a number that is easily divisible by four or eight, since there are four major divisions, or tic marks, on this axis. However, this is not mandatory. When the scales are entered, the resultant HiRes graphics histogram is drawn on the video display, as shown in Fig. 2-3B.

Table 2-1. Data for Example 2-1

Point No.	Value	Frequency
1	90	1
2	92	1
3	93	2
4	94	4
5	95	3
6	96	4
7	98	7
8	99	8
9	100	11
10	101	4
11	103	2
12	104	3
13	106	3
14	110	2
		total = 55

```
5000  REM  HISTOGRAM PLOT SUBROUTINE FOR VIDEO DISPLAY (PHISTGM)
5001  REM  X(I)=HORIZ.AXIS
5002  REM  Y(I)=VERTICAL AXIS
5003  REM  N=NO.OF DATA PTS.
5004  REM  T$=GRAPH TITLE
5005  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
5006  FOR I = 2 TO N
5007  IF (Y1 - Y(I)) <  = 0 THEN 5010
5008  Y1 = Y(I)
5009  GOTO 5012
5010  IF (Y2 - Y(I)) >  = 0 THEN 5012
5011  Y2 = Y(I)
5012  IF (X1 - X(I)) <  = 0 THEN 5015
5013  X1 = X(I)
5014  GOTO 5017
5015  IF (X2 - X(I)) >  = 0 THEN 5017
5016  X2 = X(I)
5017  NEXT I
5018  PRINT "MIN,MAX X VALUES ARE:  ";X1;"-->";X2
5019  PRINT "MIN,MAX Y VALUES ARE:  ";Y1;"-->";Y2
5020  PRINT
5021  INPUT "MIN,MAX X-AXIS SCALE: ? ";X1,X2
5022  INPUT "MIN,MAX Y-AXIS SCALE: ? ";Y1,Y2
5023  XD = X2 - X1:YD = Y2 - Y1
5024  REM  PLOT AXES WITH SCALES
5025  IF YD = 0 THEN YD = 1
5026  IF XD = 0 THEN XD = 1
5027  HCOLOR= 7: HGR
5028  REM  VERTICAL AXIS ------
5029  HPLOT 8,0 TO 8,152
5030  FOR Y = 152 TO 0 STEP  - (152 / 8): HPLOT 5,Y TO 7,Y: NEXT Y
5031  FOR Y = 152 TO 0 STEP  - (152 / 4): HPLOT 2,Y TO 7,Y: NEXT Y
5032  REM  HORIZ AXIS ---------
5033  HPLOT 13,154 TO 278,154
5034  FOR X = 13 TO 278 STEP 265 / 8: HPLOT X,155 TO X,157: NEXT X
5035  FOR X = 13 TO 278 STEP 265 / 4: HPLOT X,157 TO X,159: NEXT X
5036  PRINT : VTAB 24: PRINT "Y-AXIS ";Y1;":";Y2; SPC( 4)"X-AXIS ";X1;":"
      ;X2: PRINT
5037  FOR I = 1 TO N
5038  X = 278 - ((X2 - X(I)) * 265 / XD)
5039  Y = 152 - ((Y(I) - Y1) * 152 / YD)
5040  IF X > 278 OR X < 13 THEN 5044
5041  YT = Y
5042  IF YT < 0 THEN YT = 0
5043  FOR Y = YT TO 154: HPLOT X,Y: NEXT Y: REM  154 FOR EVEN BOTTOM LINE
5044  NEXT I: INVERSE
5045  PRINT "-";T$;"-";: NORMAL : PRINT  SPC( 28 -  LEN (T$))"CONTINUE?";
      : INPUT "";A$
5046  PRINT : PRINT "ANY CHANGES IN SCALE FACTORS (Y/N) ?";: INPUT "";A$
5047  IF A$ <  > "Y" THEN  RETURN
5048  GOTO 5018
```

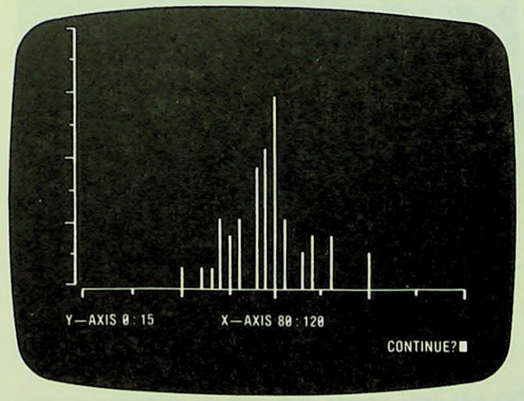
Fig. 2-1. Listing for PHISTGM subroutine.


```
100  REM  SAMPLE MAIN PROGRAM
105  DIM X(265),Y(265):SP$ = " "
110  TEXT : HOME : INPUT "NUMBER OF DATA POINTS? ";N
115  PRINT : PRINT : FOR I = 1 TO N
116  IF I > 9 THEN SP$ = ""
120  PRINT "POINT ";SP$;I;
125  INPUT "   X = ";X(I)
130  CALL  - 998
135  HTAB 26
140  INPUT "Y = ";Y(I)
145  NEXT I
150  PRINT : INPUT "TITLE? ";T$
155  PRINT : GOSUB 5000: TEXT : HOME
160  END : STOP : END : STOP
```

Fig. 2-2. Simple main program for PHISTGM.

```
NUMBER OF DATA POINTS ?  14

POINT 1   X=90        Y=1
POINT 2   X=92        Y=1
POINT 3   X=93        Y=2
POINT 4   X=94        Y=4
POINT 5   X=95        Y=3
POINT 6   X=96        Y=4
POINT 7   X=98        Y=7
POINT 8   X=99        Y=8
POINT 9   X=100       Y=11
POINT 10  X=101       Y=4
POINT 11  X=103       Y=2
POINT 12  X=104       Y=3
POINT 13  X=106       Y=3
POINT 14  X=110       Y=2

TITLE ?  100 OHM RESISTOR SAMPLE

MIN,MAX X VALUES ARE:  90-->110
MIN,MAX Y VALUES ARE:  1-->11

MIN,MAX X-AXIS SCALE:  ?  80,120
MIN,MAX Y-AXIS SCALE:  ?  0,15
```

(A) Output results.

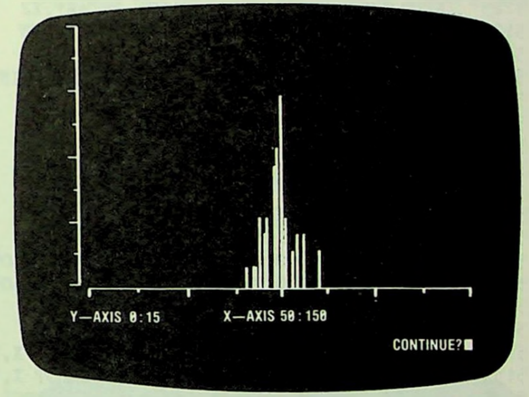


(B) Video plot.

Fig. 2-3. Plot of resistor histogram.

```
ANY CHANGE IN SCALE FACTORS (Y/N)  ?  Y

MIN,MAX X VALUES ARE:  80-->120
MIN,MAX Y VALUES ARE:  0-->15

MIN,MAX X-AXIS SCALE:  ?  50,150
MIN,MAX Y-AXIS SCALE:  ?  0,15
```

(A) Entries for new scales.



(B) Video plot.

Fig. 2-4. Histogram of resistors with new scales.

The histogram then remains on the screen until the RETURN key is pressed, after which we are asked whether or not we are satisfied with how the axes were scaled, as shown by the new entries in Fig. 2-4A and the resultant graph in Fig. 2-4B. The process of changing the axes scales can be repeated as many times as desired.

CARTESIAN PLOTS

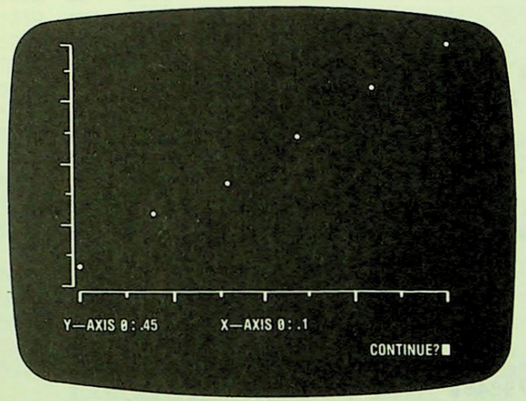
The program for generating a Cartesian, or rectangular X-Y coordinate, plot is similar to the PHISTGM subroutine. The PCARTXY subroutine is shown in Fig. 2-5, and can be used with the

```
5100  REM  CARTESIAN X-Y PLOT SUBROUTINE FOR VIDEO DISPLAY (PCARTXY)
5101  REM  HOWARD M. BERLIN LTD.
5102  REM  X(I)=HORIZ.AXIS
5103  REM  Y(I)=VERTICAL AXIS
5104  REM  N=NO.OF DATA PTS.
5105  REM  T$=GRAPH TITLE
5106  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
5107  FOR I = 2 TO N
5108  IF (Y1 - Y(I)) <  = 0 THEN 5111
5109  Y1 = Y(I)
5110  GOTO 5113
5111  IF (Y2 - Y(I)) >  = 0 THEN 5113
5112  Y2 = Y(I)
5113  IF (X1 - X(I)) <  = 0 THEN 5116
5114  X1 = X(I)
5115  GOTO 5118
5116  IF (X2 - X(I)) >  = 0 THEN 5118
5117  X2 = X(I)
5118  NEXT I
5119  PRINT
5120  PRINT "MIN,MAX X VALUES ARE:  ";X1;"-->";X2
5121  PRINT "MIN,MAX Y VALUES ARE:  ";Y1;"-->";Y2
5122  PRINT
5123  INPUT "MIN,MAX X-AXIS SCALE: ? ";X1,X2
5124  INPUT "MIN,MAX Y-AXIS SCALE: ? ";Y1,Y2
5125  XD = X2 - X1:YD = Y2 - Y1
5126  HOME : VTAB 24: INPUT "FILL BETWEEN POINTS ? ";FILL$
5127  PRINT : PRINT
5128  IF XD = 0 THEN XD = 1
5129  IF YD = 0 THEN YD = 1
5130  REM   PLOT AXES WITH SCALES
5131  HCOLOR= 7: HGR
5132  REM  VERTICAL AXIS--------
5133  HPLOT 8,1 TO 8,150
5134  FOR Y = 150 TO 0 STEP  - (150 / 8): HPLOT 5,Y TO 7,Y: NEXT Y
5135  FOR Y = 150 TO 0 STEP  - (150 / 4): HPLOT 3,Y TO 7,Y: NEXT Y
5136  REM   HORIZ AXIS ----------
5137  HPLOT 13,154 TO 279,154
5138  FOR X = 13 TO 279 STEP 266 / 8: HPLOT X,155 TO X,157: NEXT X
5139  FOR X = 13 TO 279 STEP 266 / 4: HPLOT X,157 TO X,159: NEXT X
5140  VTAB 24: PRINT "Y-AXIS ";Y1;":";Y2; SPC( 4)"X-AXIS ";X1;":";X2: PRINT
5141  FOR I = 1 TO N
5142  X = 279 - ((X2 - X(I)) * 266 / XD)
5143  Y = 150 - ((Y(I) - Y1) * 150 / YD)
5144  IF X > 279 OR X < 13 OR Y < 0 OR Y > 154 THEN 5149
5145  IF I = 1 THEN 5148
5146  IF FILL$ <  > "Y" THEN 5148
5147  HPLOT  TO X,Y
5148  HPLOT X,Y
5149  NEXT I
5150  INVERSE
5151  PRINT "-";T$;"-";: NORMAL : PRINT  SPC( 28 -  LEN (T$))"CONTINUE?";
      : CALL  - 756
5152  PRINT : PRINT : PRINT "ANY CHANGES IN SCALE FACTORS (Y/N) ?";: INPUT
      "";A$
5153  IF A$ <  > "Y" THEN  RETURN
5154  GOTO 5120
```

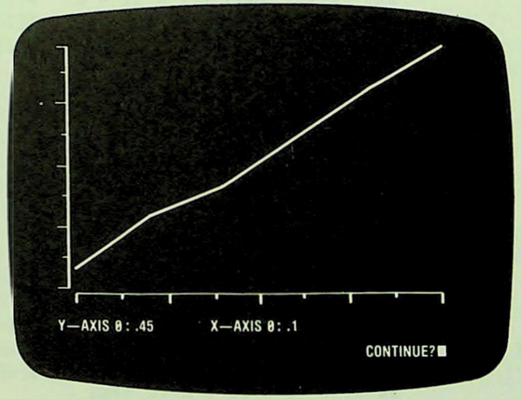
Fig. 2-5. Listing for PCARTXY subroutine.


```
NUMBER OF DATA POINTS ?  6

POINT 1    X=0          Y=.032
POINT 2    X=.02        Y=.135
POINT 3    X=.04        Y=.187
POINT 4    X=.06        Y=.268
POINT 5    X=.08        Y=.359
POINT 6    X=.1         Y=.435

TITLE ?  EXAMPLE DATA

MIN,MAX X VALUES ARE:  0-->.1
MIN,MAX Y VALUES ARE:  .032-->.435

MIN,MAX X-AXIS SCALE:  ?  0,.1
MIN,MAX Y-AXIS SCALE:  ?  0,.45

FILL BETWEEN POINTS ?  N
```

(A) Output results.



(B) Video plot—six discrete points.



(C) Video plot—points connected.

Fig. 2-6. Plotted data points.

driver program of Fig. 2-2, except that the GOSUB 5000 statement in line 155 is now changed to GOSUB 5100. When the PCARTXY subroutine is executed, you can choose to plot either the individual data points, or have the points connected by straight line segments when asked to respond to the question, FILL BETWEEN POINTS.

Example 2-2

Graph the following 6 discrete (X,Y) points: (0, 0.032), (0.02, 0.135), (0.04, 0.187), (0.06, 0.268), (0.08, 0.359), and (0.1, 0.435). Fig. 2-6A shows the results after this data is entered. The X-axis varies from 0 to 0.1 as Y ranges from 0.032 to 0.435. Fig. 2-6B shows the plot of the 6 discrete points, while Fig. 2-6C shows the same points connected together by straight line segments.

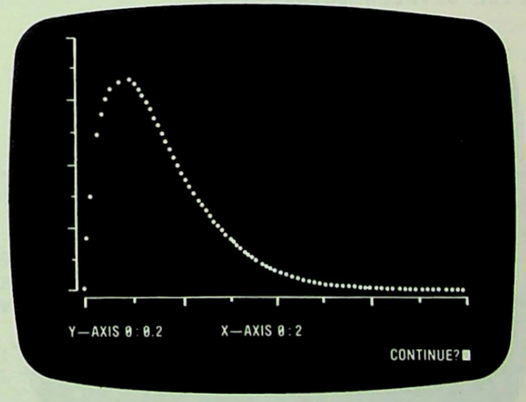
On the other hand, suppose we are interested in graphing continuous mathematical functions instead of discrete data points. The following example illustrates how this is accomplished.

Example 2-3

Suppose the transient response of a given network has the known time varying solution:

```
MIN,MAX VALUES OF T ?  0,2

TITLE ? Y=2.3*T*EXP(-5*T)

MIN,MAX X VALUES ARE:  0-->1.99245285
MIN,MAX Y VALUES ARE:  0-->.169194797

MIN,MAX X-AXIS SCALE:  ?  0,2
MIN,MAX Y-AXIS SCALE:  ?  0,.2

FILL BETWEEN POINTS ?  N
```

(A) Output results.



(B) Video plot.

Fig. 2-8. T plotted—minimum and maximum values.

```
100  REM    PCARTXY FUNCTION PLOT
101  HOME : TEST : VTAB 10: DIM X(265),Y(265)
102  N = 265 : I = 1
103  INPUT "MIN,MAX VALUES OF T ? ";A,B
104  C = (B - A) / 265
105  FOR T = A TO B STEP C
106  Y = 2.3 * T * EXP ( - 5 * T ) : REM  FUNCTION
107  X(I) = T : Y(I) = Y : I = I + 1
108  NEXT T : PRINT
109  INPUT "TITLE ? ";T$
110  GOSUB 5100 : TEXT : HOME : END
```

Fig. 2-7. Main program for plotting a function of time.

$$y(t) = 2.3te^{-5t}$$

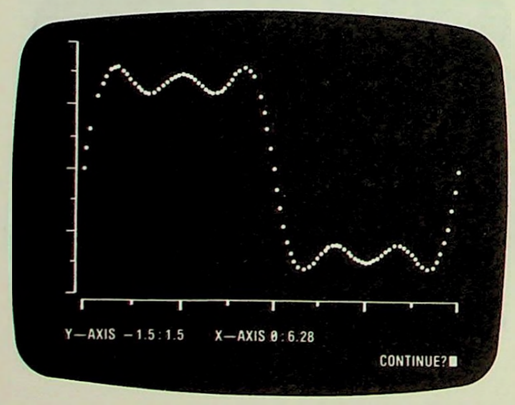
Fig. 2-7 lists the main program used to plot this function, which is defined in line 106 in terms of the variables, y, and time, t (the independent variable). Line 107 transforms these two variables to the array variables Y(I) and X(I) respectively, which are required by the PCARTXY subroutine.

```
100  REM    PCARTXY FUNCTION PLOT
101  HOME : TEST : VTAB 10: DIM X(265),Y(265)
102  N = 265 : I = 1
103  INPUT "MIN,MAX VALUES OF T ? ";A,B
104  C = (B - A) / 265
105  FOR T = A TO B STEP C
106  Y = 1.273 * SIN (T) + .424 + SIN (3 * T)
     + .255 * SIN (5 * T)
107  X(I) = T : Y(I) = Y : I = I + 1
108  NEXT T : PRINT
109  INPUT "TITLE ? ";T$
110  GOSUB 5100 : TEXT : HOME : END
```

(A) Main program.

```
MIN,MAX VALUES OF T ?  0, 6.28

TITLE ? FOURIER SQUARE WAVE

MIN,MAX X VALUES ARE:  0-->6.25630193
MIN,MAX Y VALUES ARE:  -1.18799496-->1.18798466

MIN,MAX X-AXIS SCALE:  ?  0, 6.28
MIN,MAX Y-AXIS SCALE:  ?  -1.5, 1.5

FILL BETWEEN POINTS ?  N
```

(B) Output results.



(C) Video plot.

Fig. 2-9. Plotting a function using PCARTXY.


```
5200  REM  SEMILOGARITHMIC PLOT SUBROUTINE FOR VIDEO DISPLAY (PSEMILOG)
5201  REM  X(I)=HORIZ.AXIS
5202  REM  Y(I)=VERTICAL AXIS
5203  REM  N=NO.OF DATA PTS.
5204  REM  T$=GRAPH TITLE
5205  REM  DETERMINE MINIMUM AND
5206  REM  MAXIMUM VALUES OF X & Y
5207  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1):Z9 = 1
5208  FOR I = 2 TO N
5209  IF (Y1 - Y(I)) <  = 0 THEN 5212
5210  Y1 = Y(I)
5211  GOTO 5214
5212  IF (Y2 - Y(I)) >  = 0 THEN 5214
5213  Y2 = Y(I)
5214  IF (X1 - X(I)) <  = 0 THEN 5217
5215  X1 = X(I)
5216  GOTO 5219
5217  IF (X2 - X(I)) >  = 0 THEN 5219
5218  X2 = X(I)
5219  NEXT I
5220  Y3 = Y1:Y4 = Y2: PRINT
5221  PRINT : PRINT "MIN,MAX X VALUES ARE:  ";X1;"-->";X2
5222  PRINT "MIN,MAX Y VALUES ARE:  ";Y3;"-->";Y4
5223  PRINT "MIN,MAX Y-AXIS SCALE: ? ";
5224  XY =  LOG (X1) /  LOG (10):XU =  LOG (X2) /  LOG (10):XS = X2 / X1:A
      A =  LOG (XS) /  LOG (10):AB =  INT (AA)
5225  CX =  INT ( ABS (XY - XU)) + 1
5226  IF  ABS (AE - XU) -  INT ( ABS (AE - XU)) < 1E - 4 THEN CX = CX - 1
5227  AE =  LOG (X1) /  LOG (10):AF =  INT (AE): IF (AE - AF) <  = .95 THEN
      5229
5228  XL = 10 ^ AE: GOTO 5231
5229  XL = 10 ^ AF
5230  AC =  INT (AE):XL = 10 ^ AC
5231  IF XL * (10 ^ CX) < X2 THEN CX = CX + 1
5232  IF CX > 4.9 THEN  HOME : VTAB 12: INVERSE : PRINT "DATA EXCEEDS 4 C
      YCLES CANNOT PLOT GRAPH": NORMAL : END
5233  INPUT "";Y1,Y2
5234  YD = Y2 - Y1: HOME : VTAB (24)
5235  INPUT "FILL BETWEEN POINTS? ";FILL$
5236  PRINT : PRINT
5237  REM  PLOT AXES WITH SCALES
5238  REM  ALL SAME Y OR X ?
5239  IF CX = 0 THEN CX = 1
5240  IF YD = 0 THEN YD = 1
5241  HCOLOR= 7: HGR
5242  REM  VERTICAL AXIS--------
5243  HPLOT 8,0 TO 8,150
5244  FOR Y = 150 TO 0 STEP  - (150 / 8): HPLOT 5,Y TO 7,Y: NEXT Y
5245  FOR Y = 150 TO 0 STEP  - (150 / 4): HPLOT 3,Y TO 5,Y: NEXT Y
5246  REM   HORIZ AXIS ----------
5247  HPLOT 13,154 TO 279,154
5248  U = 266 / CX: FOR X = 13 TO 279 STEP U: HPLOT X,155 TO X,160: NEXT X
5249  RT = 266 / CX:W = 0: FOR MM = 1 TO CX: FOR PZ = 2 TO 10 STEP 2
5250  LQ = (( LOG (PZ) /  LOG (10)) * RT) + 13 + W: HPLOT LQ,155 TO LQ,157
      : NEXT PZ:W = RT + W: NEXT MM
5251  VTAB 24: PRINT "MIN,MAX  Y:";Y3;",";Y4;"  X:";X1;",";X2
5252  PRINT "SCALE  Y=";Y1;":";Y2;"  X=";XL;":";XL * 10 ^ CX
```

Continued on next page.

Fig. 2-10. Listing for PSEMILOG subroutine.

```
5253  Z = 0
5254  FOR G = 0 TO CX7:H(G) = XL * 10 ^ G: NEXT G
5255  :
5256  FOR I = 1 TO N
5257  X = 13 + (( LOG (X(I) / XL)) * 266 / ( LOG (10) * CX))
5258  Y = 150 - ((Y(I) - Y1) * 150 / YD)
5259  IF X > 279 OR X < 13 OR Y < 0 OR Y > 154 THEN 5264
5260  IF I = 1 THEN 5263
5261  IF FILL$ <  > "Y" THEN 5263
5262  HPLOT  TO X,Y
5263  HPLOT X,Y
5264  NEXT I
5265  INVERSE
5266  PRINT "-";T$;"-";: NORMAL : PRINT  SPC( 28 -  LEN (T$))"CONTINUE?";
      : CALL  - 756
5267  PRINT : PRINT : PRINT "ANY CHANGES IN SCALE FACTORS (Y/N) ?";: INPUT
      "";A$
5268  IF A$ <  > "Y" THEN  RETURN
5269  GOTO 5221
```

Fig. 2-10 (cont). Listing for PSEMILOG subroutine.

Since most functions are single-valued, there will be 265 continuous points, which are recognized by the DIM statement in line 101. When this main program is run, we are first asked to select the minimum and maximum values of the independent variable (t) over which this function is to be plotted, after which we are asked for a suitable title for our graph.

For this example, we are interested in how this function appears as time varies from 0 to 2.0 seconds. When executed, we find that y ranges from 0 to 0.17 as t varies from 0 to 2.0, as shown in Fig. 2-8A, while the resulting plot is shown in Fig. 2-8B. The graph remains on the screen until the RETURN key is pressed, after which we are asked whether or not we wish to change the scale limits of either one or both axes.

Periodic, or circular functions, such as those which have sine or cosine terms are easily graphed on the display, as illustrated by the following example.

Example 2-4

Plot the first three terms for the Fourier series of a square wave, such that:

$$f(x) = 1.273\sin(x) + 0.424\sin(3x) + 0.255\sin(5x)$$

Fig. 2-9A lists the main program used, plotting the function in terms of the independent variable x, which is chosen to range from 0 to 6.28 radians. Fig. 2-9B shows the output results along with the plot of this function in Fig. 2-9C.

SEMILOGARITHMIC PLOTS

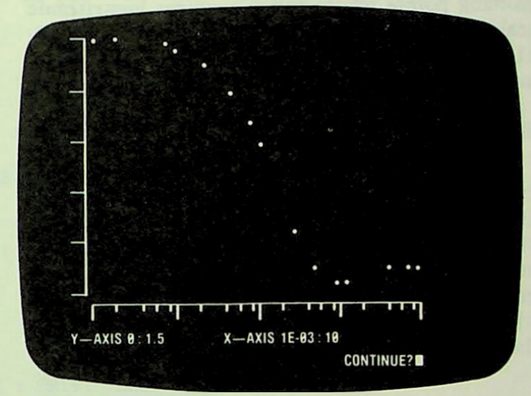
The PSEMILOG subroutine, listed in Fig. 2-10, permits the plotting of points on a semilogarithmic coordinate system, which is logarithmic in X and linear in Y. Depending on the range of values for the horizontal axis, PSEMILOG automatically scales the X-axis for values with a maximum of

```
100  REM  SAMPLE MAIN PROGRAM
105  DIM X(265),Y(265):SP$ = " "
110  TEXT : HOME : INPUT "NUMBER OF DATA POINTS? ";N
115  PRINT : PRINT : FOR I = 1 TO N
116  IF I > 9 THEN SP$ = ""
120  PRINT "POINT ";SP$;I;
125  INPUT "   X = ";X(I)
130  CALL  - 998
135  HTAB 26
140  INPUT "Y = ";Y(I)
145  NEXT I
150  PRINT : INPUT "TITLE? ";T$
155  PRINT : GOSUB 5200: TEXT : HOME
160  END : STOP : END : STOP
```

Fig. 2-11. Simple main program for PSEMILOG.


```
NUMBER OF DATA POINTS TO BE PLOTTED ?  15

POINT 1    X=.001      Y=1.49
POINT 2    X=.002      Y=1.48
POINT 3    X=.008      Y=1.43
POINT 4    X=.01       Y=1.40
POINT 5    X=.025      Y=1.29
POINT 6    X=.05       Y=1.15
POINT 7    X=.088      Y=0.95
POINT 8    X=.11       Y=.85
POINT 9    X=.3        Y=.33
POINT 10   X=.5        Y=.12
POINT 11   X=.94       Y=.013
POINT 12   X=1.23      Y=.01
POINT 13   X=4.3       Y=.099
POINT 14   X=7.1       Y=.098
POINT 15   X=9.2       Y=.098

MIN,MAX X VALUES ARE:  1E-3-->9.2
MIN,MAX Y VALUES ARE:  .01-->1.49

MIN,MAX Y-AXIS SCALE:  ?  0,1.5
```

(A) Output results.



(B) Video plot.

Fig. 2-12. Number of data points plotted.

4 cycles. However, the user is still able to set the range for the Y-axis scale.

Example 2-5

Plot the following 15 X-Y data points: (0.001, 1.49), (0.002, 1.48), (0.008, 1.43), (0.01, 1.40), (0.025, 1.29), (0.05, 1.15), (0.088, 0.95), (0.11, 0.85), (0.3, 0.33), (0.5, 0.12), (0.94, 0.013), (1.23, 0.01), (4.3, 0.099), (7.1, 0.098), and (9.2, 0.098). Using the driver program of Fig. 2-11, the resulting scale limits and graph for these 15 points are shown in Fig. 2-12. The graph, which shows 4 cycles, remains on the screen until the RETURN key is pressed. Like the PCARTXY subroutine, PSEMILOG allows you to connect straight line segments between consecutive points.

Continuous functions that require a semilogarithmic coordinate system (such as magnitude and phase plots of transfer functions, or Bode plots) are also easily graphed using PSEMILOG.

Fig. 2-13 lists a main program that can be used with PSEMILOG to plot, for example, the magnitude response:

$$dB = -10\log(1+\omega^4)$$

The magnitude function is defined in line 108. Line 109 transforms the variables T (i.e., ω) and Y to the arrays X(I) and Y(I) respectively which are required by the PSEMILOG subroutine.

Fig. 2-14 shows the magnitude plot for the transfer function as frequency is chosen to vary

```
100  REM    SAMPLE MAIN PROGRAM
101  HOME : TEST : VTAB 10: DIM X(265),Y(265)
102 N = 265 : I = 1
103  INPUT "MIN,MAX VALUES OF T ? ";A,B
104 D = LOG (B / A) / LOG (10)
105 D = INT (D + .5)
106  FOR F = 0 TO D - 1
107  FOR T = A * 10 ^ F TO (B / 10 ^  (D - F - 1)) +
     A / 10 ^ F STEP (A * 10 ^ F ) * (D) / 10
108 Y = - 10 * LOG (1 + (T ^ 4)) / LOG (10)
109 X(I) = T : Y(I) = Y : I = I + 1
110  NEXT T
111  NEXT F
112  N = N - 1
113  GOSUB 5200 : TEXT : HOME : END
```

Fig. 2-13. Simple main program for plotting magnitude using PSEMILOG.

from 0.1 to 10 Hz, or a range of two decades (cycles).

LOG-LOG PLOTS

Fig. 2-15 lists the LOGLOG subroutine used for plotting points whose range requires logarithmic scales for both axes. The main program of Fig. 2-11 can be used except that the GOSUB 5200 statement is changed to GOSUB 5300 in line 155.

Example 2-6

Graph the following 5 discrete (X,Y) points: (0.00075, 0.457), (0.003, 0.349), (0.008, 0.191), (0.015, 0.083), and (0.052, 0.01). Using the driver program of Fig. 2-11 and the GOSUB 5300 statement in line 155, the resulting scale limits and graph for these 5 points are shown in Fig. 2-16 which shows that the X-axis has 3 cycles while the Y-axis has 2 cycles.

POLAR PLOTS

Polar plots are frequently used in field theory and control systems, where the plot describes both a radial distance as well as the angle (as measured counterclockwise from the horizontal) associated with it. Examples are the electric field distribution of an antenna, or Nyquist plots of the stability of feedback systems.

The VPOLAR program shown in Fig. 2-17 permits the plotting of either discrete data points or functions expressed in polar coordinates. The polar function must be defined in line 103. The variable I is in degrees and is converted to radians by dividing it by 57.29577. As examples, Table 2-2 lists the polar forms of a variety of frequently used functions, where the angle is expressed in radians.

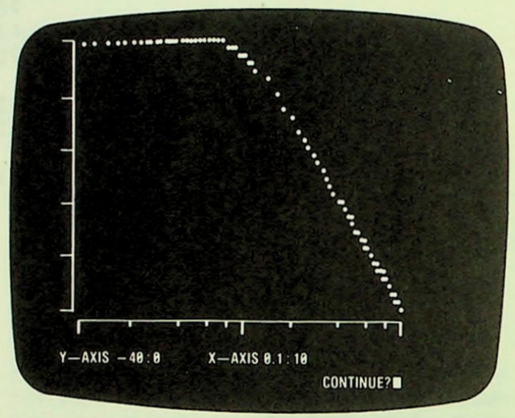


Fig. 2-14. Magnitude plot for the transfer function.

After the data points are entered or the polar function is determined, the program asks for a graph title, and then determines the maximum radius that exists for the discrete data points entered or the function that was defined. We are then asked to enter a value for the radius which is the maximum scales for both axes. The plot is then shown on the video display. In addition, the title, maximum radius, X-axis and Y-axis increments are displayed. In line 105 the variable RAD is defined which is used to represent the angle in terms of radians.

Table 2-2. Common Polar Functions

Name	Equation
Circle:	$r = A\cos\theta$ or $r = \lvert A\sin\theta\rvert$
Cardioid:	$r = A(1+\cos\theta)$ or $r = \lvert A(1+\sin\theta)\rvert$
Lemninscate:	$r = \lvert A\cos(n\theta)\rvert$ or $r = \lvert A\sin(n\theta)\rvert$
Limacon:	$r = \lvert A + B\cos\theta\rvert$ or $r = \lvert A + B\sin\theta\rvert$
N-Leaved Rose:	$r = \lvert A\cos(n\theta)\rvert$ or $r = \lvert A\sin(n\theta)\rvert$
Archimedian Spiral:	$r = A\theta$
Hyperbolic Spiral:	$r = A/\theta$
Logarithmic Spiral:	$r = \exp(b\theta)$


```
5300  REM  LOG-LOG PLOT SUBROUTINE FOR VIDEO DISPLAY (LOGLOG)
5301  REM  X(I)=HORIZ.AXIS
5302  REM  Y(I)=VERTICAL AXIS
5303  REM  N=NO.OF DATA PTS.
5304  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
5305  FOR I = 2 TO N
5306  IF (Y1 - Y(I)) <  = 0 THEN 5309
5307  Y1 = Y(I)
5308  GOTO 5311
5309  IF (Y2 - Y(I)) >  = 0 THEN 5311
5310  Y2 = Y(I)
5311  IF (X1 - X(I)) <  = 0 THEN 5314
5312  X1 = X(I)
5313  GOTO 5316
5314  IF (X2 - X(I)) >  = 0 THEN 5316
5315  X2 = X(I)
5316  NEXT I
5317  XS = X2 / X1:AA =  LOG (XS) /  LOG (10):AB =  INT (AA):XY =  LOG (X1
      ) /  LOG (10):XU =  LOG (X2) /  LOG (10):YY =  LOG (Y1) /  LOG (10):
      YU =  LOG (Y2) /  LOG (10)
5318  CX =  INT ( ABS (XY - XU)) + 1
5319  IF  ABS (AE - XU) -  INT ( ABS (AE - XU)) < 1E - 4 THEN CX = CX - 1
5320  AE =  LOG (X1) /  LOG (10):AF =  INT (AE): IF (AE - AF) <  = .95 THEN
      5322
5321  XL = 10 ^ AE: GOTO 5324
5322  XL = 10 ^ AF
5323  AC =  INT ( LOG (X1) /  LOG (10)):XL = 10 ^ AC
5324  IF XL * (10 ^ CX) < X2 THEN CX = CX + 1
5325  IF CX > 4.9 THEN  HOME : VTAB 12: INVERSE : PRINT "DATA EXCEEDS 4 C
      YCLES. CANNOT PLOT GRAPH": NORMAL : END
5326  YS = Y2 / Y1:BA =  LOG (YS) /  LOG (10):BB =  INT (BA)
5327  CY =  INT ( ABS (YY - YU)) + 1
5328  IF  ABS (BE - YU) -  INT ( ABS (BE - YU)) < 1E - 4 THEN CY = CY - 1
5329  BE =  LOG (Y1) /  LOG (10):BF =  INT (BE): IF (BE - BF) <  = .95 THEN
      5331
5330  YL = 10 ^ BE: GOTO 5333
5331  YL = 10 ^ BF
5332  BC =  INT (BE):YL = 10 ^ BC
5333  IF YL * (10 ^ CY) < Y2 THEN CY = CY + 1
5334  IF CY > 4.9 THEN  HOME : VTAB 12: INVERSE : PRINT "DATA EXCEEDS 4 C
      YCLES. CANNOT PLOT GRAPH": NORMAL : END
5335  HOME : VTAB 24: INPUT "FILL BETWEEN POINTS ? ";FILL$
5336  PRINT : PRINT
5337  REM  ALL SAME Y,X ?
5338  IF CX = 0 THEN CX = 1
5339  IF CY = 0 THEN CY = 1
5340  REM  PLOT AXES WITH SCALES
5341  HCOLOR= 7: HGR
5342  REM  VERTICAL AXIS--------
5343  HPLOT 8,0 TO 8,150
5344  FOR Y = 150 TO 0 STEP  - (150 / CY): HPLOT 3,Y TO 7,Y: NEXT Y
5345  RT = 150 / CY:W = 0
5346  FOR MM = 1 TO CY: FOR PZ = 2 TO 10 STEP 2:LQ = 150 - (( LOG (PZ) /
      LOG (10)) * RT) + W: HPLOT 5,LQ TO 6,LQ: NEXT PZ:W = W - RT: NEXT M
      M
5347  REM   HORIZ AXIS ----------
5348  HPLOT 13,154 TO 279,154
5349  U = 266 / CX: FOR X = 13 TO 279 STEP U: HPLOT X,155 TO X,160: NEXT X
```

Continued on next page.

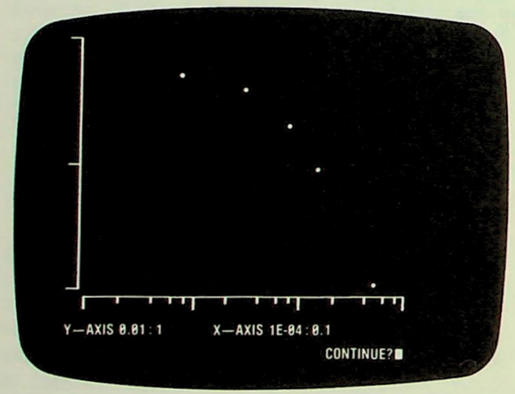
Fig. 2-15. Listing for LOGLOG subroutine.

```
5350  RT = 266 / CX:W = 0
5351  FOR MM = 1 TO CX: FOR PZ = 2 TO 10 STEP 2:LQ = (( LOG (PZ) /  LOG (
      10)) * RT) + 13 + W: HPLOT LQ,155 TO LQ,157: NEXT PZ:W = RT + W: NEXT
      MM
5352  :
5353  VTAB 24: PRINT "MIN,MAX  Y:";Y1;",";Y2;"  X:";X1;",";X2
5354  PRINT "SCALE  Y=";YL;":";YL * 10 ^ CY;"  X=";XL;":";XL * 10 ^ CX
5355  FOR I = 1 TO N
5356  X = 13 + (( LOG (X(I) / XL)) * 266 / ( LOG (10) * CX))
5357  Y = 150 - (( LOG (Y(I) / YL)) * 150 / (CY *  LOG (10)))
5358  IF X > 279 OR X < 13 OR Y < 0 OR Y > 154 THEN 5363
5359  IF I = 1 THEN 5362
5360  IF FILL$ <  > "Y" THEN 5362
5361  HPLOT  TO X,Y
5362  HPLOT X,Y
5363  NEXT I
5364  INVERSE
5365  PRINT "-";T$;"-";: NORMAL : PRINT  SPC( 28 -  LEN (T$))"CONTINUE?";
      : CALL  - 756
5366  PRINT : RETURN
```

Fig. 2-15 (cont). Listing for LOGLOG subroutine.

```
NUMBER OF DATA POINTS TO BE PLOTTED ?  5

POINT 1    X=.00075    Y=.457
POINT 2    X=.003      Y=.349
POINT 3    X=.008      Y=.191
POINT 4    X=.015      Y=.083
POINT 5    X=.052      Y=.01

MIN,MAX X VALUES ARE:  7.5E-04-->.052
MIN,MAX Y VALUES ARE:  .01-->.457
```

(A) Output results.



(B) Video plot.

Fig. 2-16. Five data points plotted.

Example 2-7

Plot the cardioid

$$r = 2(1 + \cos(\theta))$$

so that there will be a point plotted every 4 degrees. Line 103 should then read:

```
103 R(K) = 2 * (1 + COS (I / RAD))
```

When run, the maximum radius (or the maximum value of the function) was found to be 4.0, giving the results shown in Fig. 2-18. Other polar func-


```
100  REM  POLAR PLOT FOR VIDEO DISPLAY (VPOLAR)
101  LOMEM: 16384: GOTO 105
102  REM  -- ACTUAL FUNCTION --
103  R(K) = 2 * (1 +  COS (I / RAD))
104  RETURN
105  RAD = 57.29577: REM   RADIANS
106  TEXT
107  HOME : VTAB (6): PRINT  TAB( 4): INVERSE : PRINT "POLAR PLOTTING FOR
     VIDEO DISPLAY": NORMAL
108  VTAB 20:Z9 = 1
109  PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
110  VTAB (13): PRINT  SPC( 11)"APPLESOFT VERSION"
111  FOR II = 1 TO 2500: NEXT
112  DIM X(361),Y(361),R(361)
113  HOME : PRINT : PRINT "POLAR FUNCTION IS DEFINED IN LINE 103"
114  PRINT : PRINT "FOR EXAMPLE:"
115  PRINT "103 R(K) = 2 * (1 + COS (I/RAD))"
116  PRINT : PRINT "(EXAMPLE DESCRIBES A CARDIOID)       "
117  PRINT "--> YOU MAY USE 'RAD' FOR 57.29578  "
118  VTAB (12): PRINT "GRAPH IS FOR:": PRINT
119  PRINT  SPC( 12)"1)  DISCRETE DATA POINTS": PRINT : PRINT  SPC( 12)"2
     )  A POLAR FUNCTION"
120  VTAB (23): INPUT "YOUR CHOICE? ";II
121  IF II < 1 OR AA > 2 THEN 120
122  HOME : VTAB (12): PRINT "ENTER TITLE: ";: INPUT "";T$
123  IF  LEN (T$) > 20 THEN 122
124  IF II = 2 THEN 168
125  VTAB (16): INPUT "NUMBER OF DATA POINTS ? ";N
126  PRINT : FOR I = 1 TO N: PRINT "POINT #";I;": ";: INPUT "MAGNITUDE, A
     NGLE? ";R(I),Z
127  A = Z / RAD:X(I) = R(I) *  COS (A):Y(I) = R(I) *  SIN (A): NEXT
128  Z9 = 2: GOTO 177
129  PRINT : PRINT : PRINT "DO YOU WANT ME TO FILL IN THE SPACES": PRINT
     : INPUT "BETWEEN THE POINTS (Y/N) ";FILL$
130  PRINT : PRINT : PRINT : HGR : HCOLOR= 7
131  S = 4:T = 50
132  HPLOT 140,4 TO 140,155: HPLOT 50,80 TO 229,80
133  FOR II = 1 TO 11
134  HPLOT 138,S TO 139,S:S = S + 152 / 10
135  HPLOT T,78 TO T,79:T = T + 180 / 10
136  NEXT II
137  VTAB (24): INVERSE
138  PRINT "-";T$;"-";: NORMAL : PRINT " RADIUS = ";RM
139  PRINT
140  PRINT "SCALE = "; INT ((RM / 5) * 100 + .5) / 100;" PER DIVISION ";
141  FOR K = 0 TO N
142  X = 140 + (X(K) * 89 / RM):Y = 80 - (Y(K) * 76 / RM)
143  IF X > 278 OR Y > 158 THEN 149
144  IF X < 0 OR Y < 0 THEN 149
145  IF Z9 = 2 AND K = 1 AND FILL$ = "Y" THEN 148
146  IF K = 0 THEN 148
147  IF FILL$ = "Y" THEN  HPLOT  TO X,Y: GOTO 149
148  HPLOT X,Y
149  NEXT K
150  PRINT  TAB( 30);"CONTINUE?";: CALL  - 756
151  TEXT : HOME : VTAB (8)
152  PRINT  SPC( 5)"1)  CHANGE AXIS SCALES"
153  PRINT
```

Continued on next page.

Fig. 2-17. Listing for VPOLAR program.

```
154  IF Z9 <  > 2 THEN 158
155  PRINT  SPC( 5)"2)  QUIT": PRINT : PRINT : INPUT "YOUR CHOICE? ";Z8
156  ON Z8 GOTO 161,166
157  GOTO 151
158  PRINT  SPC( 5)"2)  PLOT INCREMENT": PRINT : PRINT  SPC( 5)"3)  BOTH"
     : PRINT : PRINT  SPC( 5)"4)  QUIT": PRINT : PRINT
159  INPUT "WHICH ? ";Z7
160  PRINT : ON Z7 GOTO 161,164,165,166
161  INPUT "MAX RADIUS ";RM
162  IF RM >  = R2 THEN 129
163  INVERSE : PRINT : PRINT "PLOT RADIUS MUST BE AT LEAST ";R2: NORMAL :
     GOTO 161
164  Z9 = 0: GOTO 168
165  Z9 = 1: GOTO 168
166  PRINT
167  PRINT : END
168  K = 0
169  PRINT : INPUT "ENTER THE PLOT INCREMENT (1 TO 360) ?";G
170  HOME : VTAB (12): PRINT  TAB( 12)"COMPUTING DEGREES  ":
171  FOR I = 0 TO 361 STEP G: VTAB (14): PRINT  TAB( 20)I
172  GOSUB 103
173  IF K > 360 / G THEN 176
174  A = I / RAD:X(K) = R(K) *  COS (A):Y(K) = R(K) *  SIN (A)
175  K = K + 1: NEXT I
176  N = K - 1: REM  MIN/MAX
177  R1 = R(1):R2 = R(1)
178  FOR I = 2 TO N
179  IF (R1 - R(I)) <  = 0 THEN 181
180  R1 = R(I): GOTO 183
181  IF (R2 - R(I)) >  = 0 THEN 183
182  R2 = R(I): IF R2 < 1 THEN R2 = 1
183  NEXT I
184  IF Z9 = 0 THEN 129
185  HOME
186  PRINT "THE DATA HAS A MAXIMUM RADIUS OF ";R2
187  PRINT
188  INPUT "DESIRED MAX RADIUS FOR THE AXES ? ";RM
189  IF RM >  = R2 THEN 191
190  PRINT : INVERSE : PRINT "PLOT RADIUS MUST BE AT LEAST ";R2: NORMAL :
     PRINT : GOTO 188
191  GOTO 129
```

Fig. 2-17 (cont). Listing for VPOLAR program.

tions plotted using the VPOLAR routine are shown in Fig. 2-19.

PLOTTING WITH A LINE PRINTER

For those of you who have a line printer, you would probably like to make a hard-copy record of a particular graph either for inclusion in a report, or for later comparison with other results. The following programs allow continuous functions to be graphed on virtually all serial and parallel line printers used with the Apple II. For all these programs, the printer interface card is set at SLOT = 1. If a parallel printer is used, 80 columns maximum will be printed.

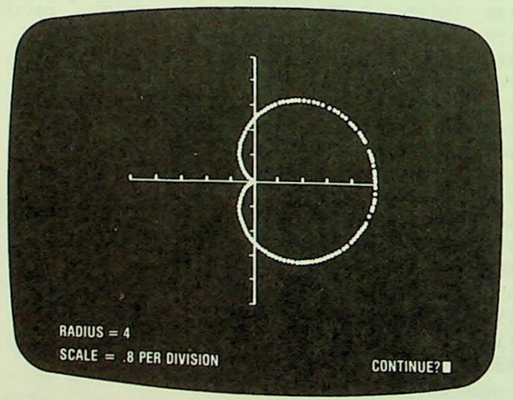
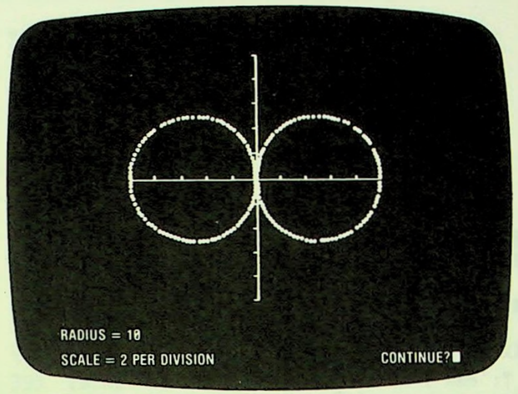
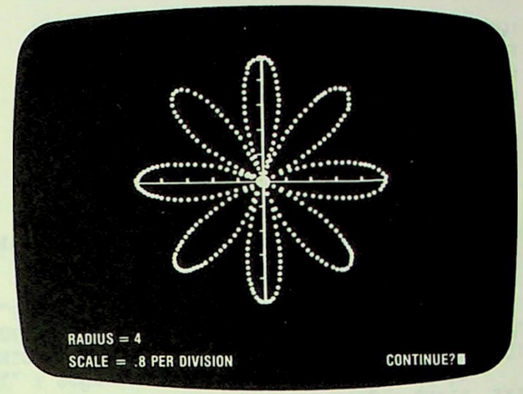


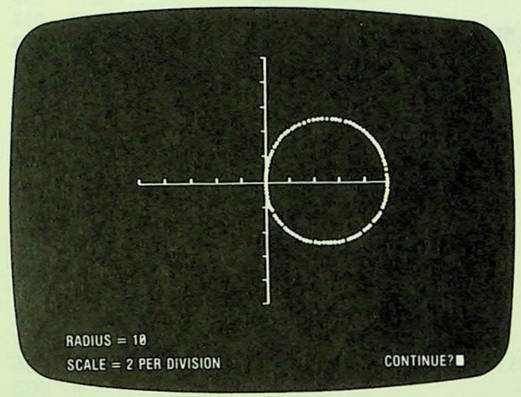
Fig. 2-18. Cardioid plot.



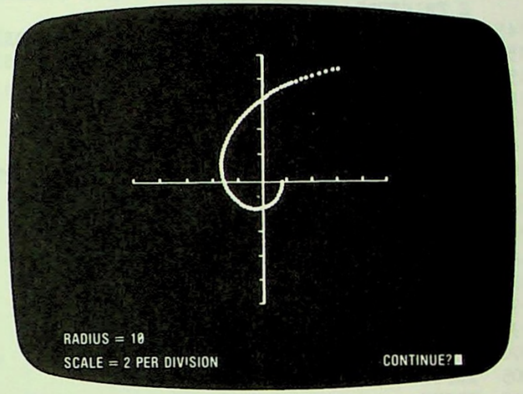
(A) Lemniscate.



(B) 6-leaved rose.



(C) Circle.



(D) Hyperbolic spiral.

(E) Archimedean spiral.

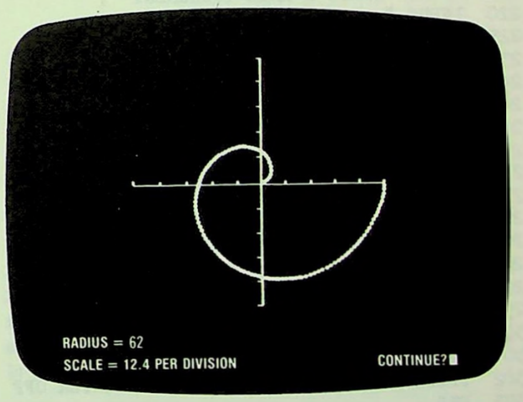


Fig. 2-19. Video plot of various polar functions.

```
100  REM ***** LINEPLOT *****
105  REM * X-Y PLOT ROUTINE *
106  REM * FOR LINE PRINTER *
107  REM ********************
108  :
109  SLOT = 1: REM   PRINTER SLOT
110  :
111  GOTO 135
112  PRINT  CHR$ (4);"PR#";SL
113  PRINT  CHR$ (9);"80N": REM   PARALLEL CARD SET TO 80 COL.
114  RETURN : REM DEL LINE 113       IF USING SERIAL CARD
115  IF XX < 0 THEN XX =  ABS (XX):NEG$ = "-"
120  IF XX > 999999.99 THEN  PRINT "TOO LARGE": END
125  IF XX < .005 THEN Z$ = ".00":LG = 3: RETURN
130  XX =  INT (XX * 100 + .5) / 100:Z$ =  STR$ (XX + .005):Z$ =  LEFT$ (Z
     $, LEN (Z$) - 1):Z$ = NEG$ + Z$:NEG$ = "":LG =  LEN (Z$): RETURN
135  LINE$ = "+----+----+----+----+----+----+----+----+----+----+"
140  HOME : VTAB (4): HTAB (4): INVERSE : PRINT "X-Y PLOT PROGRAM FOR LIN
     E PRINTER": NORMAL
145  VTAB (7): PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
150  VTAB (10): PRINT  SPC( 11)"APPLESOFT VERSION"
155  FOR II = 1 TO 2500: NEXT II
160  DIM X(100),Y(100)
161  HOME
162  INVERSE
165  PRINT "FUNCTION MUST BE DEFINED IN LINE 275  ": PRINT "EXAMPLE>  275
     Y=EXP(-.2 * X) * SIN(X)": PRINT
170  NORMAL
175  PRINT "INPUT THE MINIMUM 'X' VALUE OF"
180  PRINT "THE FUNCTION TO BE PLOTTED ";
185  INPUT "      ";X1
190  PRINT
195  PRINT "INPUT THE MAXIMUM 'X' VALUE OF"
200  PRINT "THE FUNCTION TO BE PLOTTED ";
205  INPUT "      ";X2
210  PRINT
215  PRINT "INPUT X AXIS INCREMENT ";
220  INPUT "          ";R
225  PRINT : PRINT
230  INPUT "X AXIS TITLE? ";X$
240  PRINT
245  INPUT "Y AXIS TITLE? ";Y$
250  K =  LEN (X$)
255  I = 1:XD = X2 - X1:S = XD / R
265  FOR X = X1 TO X2 STEP R
270  REM  INSERT FUNCTION HERE
275  Y =  EXP ( - .2 * X) *  SIN (X)
280  REM  INSERT FUNCTION ABOVE
285  X(I) = X:Y(I) = Y:I = I + 1:N = I - 1
290  NEXT
295  GOSUB 310
300  GOSUB 385
301  PRINT  CHR$ (12): REM  FORM FEED
302  PRINT  CHR$ (4);"PR#0": REM  PRINTER OFF
303  END
310  REM  FIND MIN,MAX VALUE OF Y
315  Y1 = Y(1):Y2 = Y(1)
320  FOR I = 1 TO N
```

Continued on next page.

Fig. 2-20. Listing for LINEPLOT program.


```
325  IF Y1 - Y(I) <  = 0 THEN 340
330  Y1 = Y(I)
335  GOTO 350
340  IF Y2 - Y(I) >  = 0 THEN 350
345  Y2 = Y(I)
350  NEXT I
351  PRINT
355  PRINT "MINIMUM Y-AXIS VALUE = ";Y1
360  PRINT "MAXIMUM Y-AXIS VALUE = ";Y2
365  PRINT : INPUT "ENTER MINIMUM Y-AXIS VALUE ";Y1
370  INPUT "ENTER MAXIMUM Y-AXIS VALUE ";Y2
375  YD = Y2 - Y1: RETURN
380  REM  ------------------
385  REM  ### PRINTIT ###
390  HOME : VTAB (12)
395  PRINT "TURN ON PRINTER & HIT ANY KEY";: CALL  - 756
400  L = YD / 50:C1 = Y1 - L / 2:D1 = Y2 + L / 2:W = 1: HOME
405  PRINT : REM  DO NOT DELETE
410  GOSUB 112: REM START PRINTER
420  PRINT  TAB( 32)Y$: PRINT  CHR$ (10)
425  PRINT  SPC( 8)" ";
430  FOR U = 0 TO 5: PRINT  SPC( CC);U * 10 * L + Y1;
435  CC = 8
440  NEXT U: PRINT " "
445  PRINT  TAB( 11)LINE$
450  FOR I = 1 TO S + 1
455  IF Y(I) < C1 THEN 515
460  IF Y(I) >  = D1 THEN 515
465  XX = X(I): GOSUB 115
470  VLTR$ =  MID$ (X$,W,1)
475  IF VLTR$ = "" THEN VLTR$ = " "
480  PRINT VLTR$;" "; TAB( 7 - LG)Z$;
485  VV = (Y(I) - Y1) / L + 1.5
490  PRINT " I"; TAB( VV);"*"; TAB( 53 - VV);"I"
500  W = W + 1: NEXT I
505  PRINT  TAB( 11)LINE$
510  RETURN
515  XX = X(I): GOSUB 115
520  PRINT  MID$ (X$,W,1);"  ";Z$
525  PRINT  TAB( 9);"I"; TAB( 52)"I": GOTO 500
```

Fig. 2-20 (cont). Listing for LINEPLOT program.

Cartesian Plots

Fig. 2-20 shows the LINEPLOT program which permits the plotting of a function, which must be defined in line 275, on a Cartesian coordinate system. If a serial printer is used, delete line 113. The program asks for the minimum and maximum values of X for which the function is to be plotted, as well as the X-axis increment scale. In addition, the program will label both axes.

Example 2-8

Plot the function $y(t) = e^{-.2x}\sin(x)$ as x varies from 0 to 15 radians. Fig. 2-21 shows the resultant graph, for which the scale of the Y-axis was set to vary from −0.5 to 1.0.

Semilogarithmic Plots

The XLOGPLOT program shown in Fig. 2-22 permits plotting of a function having a logarithmic behavior, which must be defined in line 275. Such would be the case of the frequency response of a filter which might cover several decades. Line 435 must be deleted if a serial printer is used.

Example 2-9

Plot the frequency response of a second-order low-pass filter, having the form:

$$dB = -10\log(1 + f^2)$$

```
MINIMUM Y-AXIS VALUE = -.397434
MAXIMUM Y-AXIS VALUE =  .738963

ENTER MINIMUM Y-AXIS VALUE ? -.5
ENTER MAXIMUM Y-AXIS VALUE ? 1.0
```

(A) Output results.

```
                          AMPLITUDE

          -.5       -.2        .1        .4        .7        1
           +---------+---------+---------+---------+---------+
R    0.00 I                    *                             I
A    0.50 I                              *                   I
D    1.00 I                                        *         I
I    1.50 I                                         *        I
A    2.00 I                                   *              I
N    2.50 I                               *                  I
S    3.00 I                         *                        I
     3.50 I               *                                  I
     4.00 I          *                                       I
     4.50 I     *                                            I
     5.00 I       *                                          I
     5.50 I            *                                     I
     6.00 I                  *                               I
     6.50 I                        *                         I
     7.00 I                          *                       I
     7.50 I                           *                      I
     8.00 I                          *                       I
     8.50 I                           *                      I
     9.00 I                        *                         I
     9.50 I                    *                             I
    10.00 I                   *                              I
    10.50 I                  *                               I
    11.00 I                  *                               I
    11.50 I                   *                              I
    12.00 I                    *                             I
    12.50 I                     *                            I
    13.00 I                       *                          I
    13.50 I                       *                          I
    14.00 I                        *                         I
    14.50 I                       *                          I
    15.00 I                       *                          I
          +---------+---------+---------+---------+---------+
```

(B) Printer plot.

Fig. 2-21. Function $y(t) = e^{-.2x} \sin(x)$ plotted as x varies from 0 to 15 radians.

Fig. 2-23 shows the resultant plot from 0.1 to 10 Hz.

Polar Plots

The POLARPLT program shown in Fig. 2-24 permits the plotting of polar functions (Table 2-2), such as those describing cardioids, spirals, etc., and must be defined in line 190. Line 470 must be deleted if a serial printer is used.

When run, the program requires only a graph title and the scale limit for the two axes.

Example 2-10

Plot the N-leaved rose

$$r = \sin(3\theta)$$

As shown in Fig. 2-25, the scale limits of the two axes were arbitrarily set at 1.2.


```
100  REM  ***** XLOGPLOT ****
105  REM  ** SEMI-LOG PLOT **
106  REM  **   SUBROUTINE  **
107  REM  * FOR LINEPRINTER *
108  REM  ********************
109  :
115  SLOT = 1: REM   PRINTER SLOT
116  :
117  REM DEL LINE 435 IF YOU ARE
118  REM   USING THE SERIAL CARD
120  GOTO 145
125  IF XX < 0 THEN XX =  ABS (XX):NEG$ = "-"
130  IF XX > 999999.99 THEN  PRINT "TOO LARGE": END
135  IF XX < .005 THEN Z$ = ".00":LG = 3: RETURN
140  XX =  INT (XX * 100 + .5) / 100:Z$ =  STR$ (XX + .005):Z$ =  LEFT$ (Z
     $, LEN (Z$) - 1):Z$ = NEG$ + Z$:NEG$ = "":LG =  LEN (Z$): RETURN
145  LINE$ = "+----+----+----+----+----+----+----+----+----+----+"
150  HOME : VTAB (4): HTAB (4): INVERSE : PRINT "X-Y PLOT PROGRAM FOR LIN
     E PRINTER": NORMAL
155  VTAB (7): PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
160  VTAB (10): PRINT  SPC( 11)"APPLESOFT VERSION"
165  FOR II = 1 TO 2500: NEXT II
170  DIM X(100),Y(100)
175  HOME
180  INVERSE
185  PRINT "FUNCTION MUST BE DEFINED IN LINE 275  ": PRINT "EXAMPLE>  275
     Y=EXP(-.2 * X) * SIN(X)": PRINT
190  NORMAL
195  PRINT "INPUT THE MINIMUM 'X' VALUE OF"
200  PRINT "THE FUNCTION TO BE PLOTTED ";
205  INPUT "       ";X1
210  PRINT
215  PRINT "INPUT THE MAXIMUM 'X' VALUE OF"
220  PRINT "THE FUNCTION TO BE PLOTTED ";
225  INPUT "       ";X2
230  X1 =  LOG (X1) /  LOG (10):X2 =  LOG (X2) /  LOG (10)
235  PRINT
240  PRINT : PRINT
245  INPUT "X AXIS TITLE? ";X$
250  PRINT
255  INPUT "Y AXIS TITLE? ";Y$
260  K =  LEN (X$)
265  I = 1:XD = X2 - X1:S = XD / .1
270  FOR X = X1 TO X2 + .1 STEP .1
275  REM  INSERT FUNCTION HERE
280  Y = 20 *  LOG (1 /  SQR (1 + ((10 ^ X) ^ 2))) /  LOG (10)
285  REM  INSERT FUNCTION ABOVE
290  X(I) = X:Y(I) = Y:I = I + 1:N = I - 1
295  NEXT
300  GOSUB 320
305  GOSUB 400
310  PRINT  CHR$ (4);"PR#0": REM  PRINTER OFF
315  END
320  REM  FIND MIN,MAX VALUE OF Y
325  Y1 = Y(1):Y2 = Y(1)
330  FOR I = 1 TO N
335  IF Y1 - Y(I) <  = 0 THEN 350
340  Y1 = Y(I)
```

Continued on next page.

Fig. 2-22. Listing for XLOGPLOT program.

```
345  GOTO 360
350  IF Y2 - Y(I) >  = 0 THEN 360
355  Y2 = Y(I)
360  NEXT I
365  PRINT
370  PRINT "MINIMUM Y-AXIS VALUE = ";Y1
375  PRINT "MAXIMUM Y-AXIS VALUE = ";Y2
380  PRINT : INPUT "ENTER MINIMUM Y-AXIS VALUE ";Y1
385  INPUT "ENTER MAXIMUM Y-AXIS VALUE ";Y2
390  YD = Y2 - Y1: RETURN
395  REM  ------------------
400  REM   ### PRINT IT ###
405  HOME : VTAB (12): INVERSE
410  PRINT "TURN ON PRINTER & HIT ANY KEY";: CALL  - 756
415  NORMAL
420  L = YD / 50:C1 = Y1 - L / 2:D1 = Y2 + L / 2:W = 1: HOME
425  PRINT : REM  DO NOT DELETE
430  PRINT  CHR$ (4);"PR#"SLOT: REM   PRINTER SLOT
435  PRINT  CHR$ (9);"80N": REM  LINE WIDTH 80
440  PRINT  TAB( 32)Y$: PRINT  CHR$ (10)
445  PRINT  SPC( 8)" ";
450  FOR U = 0 TO 5: PRINT  SPC( CC);U * 10 * L + Y1;
455  CC = 8
460  NEXT U: PRINT " "
465  PRINT  TAB( 11)LINE$
470  FOR I = 1 TO S + 2
475  IF Y(I) < C1 THEN 535
480  IF Y(I) >  = D1 THEN 535
485  XX = 10 ^ X(I): GOSUB 125
490  VLTR$ =  MID$ (X$,W,1)
495  IF VLTR$ = "" THEN VLTR$ = " "
500  PRINT VLTR$;" "; TAB( 7 - LG)Z$;
505  VV = (Y(I) - Y1) / L + 1.5
510  PRINT " I"; TAB( VV);"*";
515  PRINT  TAB( 53 - VV);"I"
520  W = W + 1: NEXT I
525  PRINT  TAB( 11)LINE$
530  RETURN
535  XX = 10 ^ X(I): GOSUB 125
540  PRINT  MID$ (X$,W,1);"  ";Z$
545  PRINT  TAB( 9);"I"; TAB( 52)"I": GOTO 520
```

Fig. 2-22 (cont). Listing for XLOGPLOT program.

(A) Output results.

```
MIN, MAX X? 0.1, 10.0
MIN, MAX Y? -20, 5
X-AXIS TITLE ? LOG FREQUENCY IN HZ
Y-AXIS TITLE ? DB RESPONSE
```

(B) Printer plot.

```
                              DB RESPONSE

                  -20       -15       -10       -5        0         5
                   +----+----+----+----+----+----+----+----+----+----+
L     0.10 I                                                  *      I
O     0.13 I                                                  *      I
G     0.16 I                                                  *      I
      0.20 I                                                  *      I
F     0.25 I                                                 *       I
R     0.32 I                                                 *       I
E     0.40 I                                                 *       I
Q     0.50 I                                                *        I
U     0.63 I                                               *         I
E     0.79 I                                              *          I
N     1.00 I                                            *            I
C     1.26 I                                          *              I
Y     1.58 I                                       *                 I
      2.00 I                                    *                    I
I     2.51 I                                 *                       I
N     3.16 I                              *                          I
      3.98 I                           *                             I
H     5.01 I                        *                                I
Z     6.31 I                     *                                   I
      7.94 I                  *                                      I
     10.00 I*                                                        I
           +----+----+----+----+----+----+----+----+----+----+----+
```

Fig. 2-23. Plotted frequency response of low-pass filter.

```
100  REM  ***** POLARPLT *****
101  REM  ** POLAR PLOTTING **
102  REM  * FOR LINE PRINTER *
103  REM  **********************
104  :
105  SLOT = 1: REM  PRINTER SLOT
106  REM  DEL LINE 470 IF YOU ARE      USING A SERIAL CARD
107  REM  FUNCTION IN LINE 190 **
108  :
120  DIM X(91),Y(90),G$(36):DR = .06981317
130  N = 90
140  HOME : INPUT "SCALE LIMITS ?";Z
150  INPUT "GRAPH TITLE ?";G$
160  HOME : VTAB (11): NORMAL : PRINT  TAB( 9): FLASH : PRINT "CONVERTING
     DATA POINTS": NORMAL
170  FOR I = 1 TO N:D = DR * I
175  :
176  :
180  REM   ***** FUNCTION *****
190  F =  SIN (3 * D): REM  SAMPLE
196  :
200  X(I) =  INT (((F *  COS (D) / Z + 1) * 30) + .5)
210  Y(I) =  INT ((( - F *  SIN (D) / Z + 1) * 18) + .5)
220  NEXT
230  HOME : VTAB (12): PRINT  TAB( 11): FLASH : PRINT "SORTING DATA POINT
     S": NORMAL
235  CHECK = 0
236  NN = N - 1
237  IF NN = 0 THEN 320
240  FOR I = 1 TO NN:J = I + 1
260  IF Y(I) <  = Y(J) THEN 300
270  A = Y(I):B = X(I)
280  Y(I) = Y(J):X(I) = X(J)
290  Y(J) = A:X(J) = B
295  CHECK = 1
300  NEXT I
320  G$(18) = "----------------------------I----------------------------
     "
330  T1$ = "                            I                            "
340  NORMAL : HOME : VTAB (12): PRINT  TAB( 15): FLASH : PRINT "FORMATTIN
     G": NORMAL
350  FOR I = 1 TO 17
360  G$(I) = T1$
370  NEXT I
380  FOR I = 19 TO 34
390  G$(I) = T1$
400  NEXT I
410  FOR I = 1 TO N - 1
420  IF X(I) > 58 OR Y(I) > 34 THEN 450
430  IF X(I) < 1 OR Y(I) < 1 THEN 450
440  G$(Y(I)) =  LEFT$ (G$(Y(I)),X(I) - 1) + "*" +  RIGHT$ (G$(Y(I)),59 -
     X(I))
450  NEXT I
455  VTAB (12): PRINT "          PRINTING......"
460  PRINT  CHR$ (4);"PR#";SL
470  PRINT  CHR$ (9);"80N"
480  PRINT G$
490  PRINT  TAB( 27);"+Y = ";Z
```

Continued on next page.

Fig. 2-24. Listing for POLARPLT program.


```
500  PRINT " "
510  FOR I = 1 TO 17
520  PRINT G$(I)
530  NEXT I
540  PRINT G$(18);"  +X = ";Z
550  FOR I = 19 TO 35
560  PRINT G$(I)
570  NEXT I
580  PRINT " "
590  PRINT  TAB( 27);"-Y = ";Z
600  FOR I = 1 TO 27: PRINT " ": NEXT I
610  PRINT  CHR$ (4);"PR#0"
615  PRINT  CHR$ (4);"PR#0"
620  HOME
625  VTAB (12): PRINT "          COMPLETED ....        "
630  END
```

Fig. 2-24 (cont). Listing for POLARPLT program.

```
          N-LEAVED ROSE  F=SIN(3*D)
                                                +Y = 1.2

                                                  I
                                                  I
                                                  I
```

3

The Analysis of Signals

This chapter discusses programs which are used to analyze both periodic and aperiodic (nonrepetitive) waveforms. In particular, we will be able to determine both the average and root mean square (rms) values of a periodic function, the representation of any periodic signal by Fourier series, the representation of aperiodic signals by a modification of the Fourier transform, and analyze damped oscillations.

THE AVERAGE AND RMS VALUES OF A WAVEFORM

The RMSAV program listing shown in Fig. 3-1 determines both the average and rms values of a periodic signal. This is extremely useful in verifying ac measurements, such as determining the average power of a circuit, or of a particular device within the circuit. Mathematically, the average value of a periodic waveform f(t) is:

$$AV = \frac{1}{T}\int_0^T f(t)\ dt \qquad \text{(Eq. 3-1)}$$

However, it must be noted that the average value for most periodic functions having symmetry about zero amplitude (i.e., no dc term) is naturally zero, since the same amount of the waveform is above as well as below the zero reference, like most sine, triangular, and square waves. When this occurs, a half-cycle average value is determined to represent the average value of a periodic function, so that

$$AV = \frac{2}{T}\int_0^{T/2} f(t)\ dt \qquad \text{(Eq. 3-2)}$$

On the other hand, the rms value is given by:

$$RMS = \left(\frac{1}{T}\int_0^T f(t)^2\ dt\right)^{1/2} \qquad \text{(Eq. 3-3)}$$

Using Simpson's rule to numerically compute the integrals, the RMSAV program determines the average and rms values from equally spaced discrete data points, or from a periodic function, which must be supplied in line 124, so as the argument of the periodic function must have the form:

argument = (A∗I)/57.92577

to convert the angle from degrees to radians. If the average value is zero, then the half-cycle average is determined. Furthermore, Simpson's rule requires that the number of data points be an odd number. In addition, careful attention must be paid to discontinuities which occur during the cycle, as will be illustrated in a later example.

Example 3-1

Determine the average and rms values of an ordinary 60-Hz sine wave whose amplitude varies between −1.0 and +1.0 volts. Fig. 3-2 shows the output for the RMSAV program, when the sine wave function is supplied in line 124, giving 51 data points for one complete cycle. Since the full cycle average value for this sine wave is zero, the program then computes the half-cycle average in addition to the rms value. Using classical methods, the half-cycle average and rms values for a sine wave are 0.637 and 0.707 times the peak voltage.

Example 3-2

Consider the 50% duty cycle square wave, shown in Fig. 3-3, which varies between 0 and 5 volts. We will first run the RMSAV program by entering discrete data points instead of describing

```
100  REM  RMS AND AVG VALUE OF A PERIODIC WAVEFORM (RMSAV)
101  CALL  - 936: VTAB (5): PRINT  TAB( 8): INVERSE : PRINT "RMS AND AVER
     AGE VALUE OF": NORMAL
102  VTAB (7): PRINT  TAB( 11): INVERSE : PRINT "A PERIODIC WAVEFORM": NORMAL
103  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980"
104  PRINT : PRINT  TAB( 9)"HOWARD M. BERLIN, LTD."
105  PRINT : PRINT  TAB( 16)"APPLE II"
106  PRINT : PRINT  TAB( 11)"APPLESOFT VERSION"
107  FOR I = 1 TO 2500: NEXT I: CALL  - 936
108  CLEAR : DIM C(361):RAD = 57.29578
109  PRINT : PRINT "RMS AND AVERAGE VALUES ARE TO BE": PRINT "DETERMINED
     FROM:": PRINT : PRINT  TAB( 7)"1. DISCRETE POINTS": PRINT : PRINT  TAB(
     7)"2. A DEFINED FUNCTION"
110  PRINT : PRINT : PRINT "FUNCTION MUST BE DEFINED IN LINE 124.": PRINT
     : PRINT  TAB( 14)"FOR EXAMPLE:": PRINT : PRINT : PRINT "124 C(I)=SIN
     ((A*I)/57.29577)"
111  PRINT : PRINT : INPUT "SELECT CHOICE? ";Z9
112  PRINT : INPUT "CYCLE PERIOD? ";T
113  PRINT : PRINT "NUMBER OF DIVISIONS FOR 1 CYCLE": INPUT "  (MUST BE A
     N INTEGER)  -->  ? ";C
114  V = T / C: IF Z9 <  > 1 THEN 122
116  CALL  - 936: PRINT "TIME"
117  FOR I = 0 TO C
118  PRINT I * V;" ";
119  INPUT C(I)
120  NEXT I
121  CALL  - 936: GOTO 126
122  A = 360 / C: HOME : GOSUB 150
123  FOR I = 0 TO C
124  C(I) =  SIN ((A * I) / RAD)
125  NEXT I
126  Z = 1 / T
127  S = 0:AV = 0:HC = 0
129  FOR I = 0 TO C - 2 STEP 2
130  S = S + C(I) ^ 2 + 4 * C(I + 1) ^ 2 + C(I + 2) ^ 2
131  AV = AV + C(I) + 4 * C(I + 1) + C(I + 2)
132  IF I <  = C / 2 THEN  GOSUB 143
133  NEXT I
134  AV = Z * V * AV / 3: IF AV < 1E - 3 THEN AV = 0
135  HC = Z * V * HC * 2 / 3
136  RS =  SQR (Z * V * S / 3)
137  CALL  - 936: VTAB 8: PRINT "RMS VALUE = ";RS
138  IF AV = 0 THEN 141
139  PRINT : PRINT "AVERAGE VALUE = ";AV
140  IF AV <  > 0 THEN 142
141  PRINT : PRINT "HALF CYCLE AVG. VALUE = ";HC
142  END
143  HC = HC + C(I) + 4 * C(I + 1) + C(I + 2): RETURN
150  CALL  - 936: VTAB (10): PRINT  TAB( 15): FLASH : PRINT "COMPUTING": NORMAL : RETURN
```

Fig. 3-1. Listing for RMSAV program.

the function. Square waves are unusual functions in that they have discontinuities. In order to minimize computational error, it is best to assign a value that is midway between the minimum and maximum values of the square wave (i.e., 2.5 volts) as the value of the point where the discontinuity occurs during the cycle. For the situation where the square wave is discontinuous at both the beginning and the end of the cycle, we use the values of 5 and 0 volts respectively. When numerical integration methods are used, greater accuracy is achieved when the number of points is made as large as possible. Fig. 3-4 shows the results when 21 data points are used (20 cycle divisions plus the value at t = 0).

Using classical methods to check our results,


```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 2

CYCLE PERIOD? 0.16667

NUMBER OF DIVISIONS FOR 1 CYCLE
   (MUST BE AN EVEN NUMBER) ?50

RMS VALUE = .707106723

HALF CYCLE AVERAGE VALUE = .634110557
```

Fig. 3-2. Output for RMSAV program.

we would find that the average value of such a waveform is one-half its peak value, or 2.5, while the rms value is 0.707 times the peak, or 3.535. Increasing the number of data points usually leads to greater accuracy.

However, suppose we had described the square wave mathematically instead of by discrete points. Unfortunately, we still have difficulty in handling the discontinuity at t = 0.5 second. One way around this problem is to run the program twice and then average the results. In the original RMSAV listing, lines 123–125 are replaced by the following statements:

```
123 FOR I=0 TO (C/2)-1: C(I)=5: NEXT I
124 C(C/2)=5
125 FOR I=(C/2)+1 TO C: C(I)=0: NEXT I
```

so that the results of the first run will be slightly higher than the expected value, as shown in Fig. 3-5A. Then replace line 124 by C(C/2) = 0 so that the results of the second run will be slightly lower than the expected values, as shown in Fig. 3-5B. The two results are then averaged, in which case the average value is the average of 2.51666667 and 2.48333333, or 2.5, while the rms value is

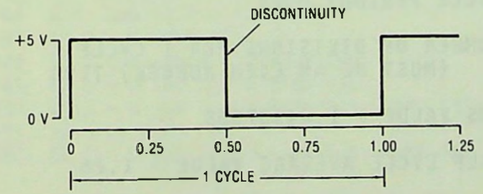


Fig. 3-3. 50% duty cycle square wave.

```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 1

CYCLE PERIOD? 1.0

NUMBER OF DIVISIONS FOR 1 CYCLE
   (MUST BE AN EVEN NUMBER) ?20

TIME
0 ?5
.05 ?5
.1 ?5
.15 ?5
.2 ?5
.25 ?5
.3 ?5
.35 ?5
.4 ?5
.45 ?5
.5 ?2.5
.55 ?0
.6 ?0
.65 ?0
.7 ?0
.75 ?0
.8 ?0
.85 ?0
.9 ?0
.95 ?0
1 ?0

RMS VALUE = 3.50594733

AVERAGE VALUE = 2.5
```

Fig. 3-4. Results when 21 data points are used.

the average of 3.54729944 and 3.52372909, or 3.535514265.

Example 3-3

Determine the average and rms values for the sawtooth waveform shown in Fig. 3-6. Unlike the square wave which has a discontinuity in the middle as well as at the beginning and end of the cycle, the sawtooth is discontinuous only at the beginning and end of the cycle.

```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 2

CYCLE PERIOD? 1.0

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?100

RMS VALUE = 3.54729944

HALF CYCLE AVERAGE VALUE = 2.51666667
```

(A) Overestimate.

```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 2

CYCLE PERIOD? 1.0

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?100

RMS VALUE = 3.52372909

HALF CYCLE AVERAGE VALUE = 2.48333333
```

(B) Underestimate.

Fig. 3-5. Rms and average values of a periodic waveform.

Fig. 3-7 shows the results using 11 data points, with no corrections made for the pair of discontinuities. The starting point is zero, while the last

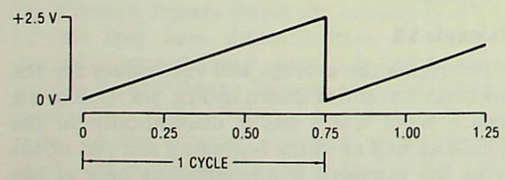


Fig. 3-6. Sawtooth waveform.


```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 1

CYCLE PERIOD? 0.75

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?10

TIME
0 ?5
.075 ?.25
.15 ?.5
.225 ?.75
.3 ? 1.0
.375 ?1.25
.45 ?1.5
.525 ?1.75
.6 ?2.0
.675 ?2.25
.75 ?2.5


RMS VALUE = 1.44337567

AVERAGE VALUE = 1.25
```

Fig. 3-7. Results when 11 data points are used.

```
RMS AND AVERAGE VALUES ARE TO BE
DETERMINED FROM:

      1. DISCRETE POINTS
      2. A DEFINED FUNCTION

FUNCTION MUST BE DEFINED IN LINE 124.
            FOR EXAMPLE:

124 C(I)=SIN((A*I)/57.29577)

SELECT CHOICE? 2

CYCLE PERIOD? 0.75

NUMBER OF DIVISIONS FOR 1 CYCLE
    (MUST BE AN EVEN NUMBER) ?100

RMS VALUE = 1.44337568

HALF CYCLE AVERAGE VALUE = 1.25
```

Fig. 3-8. Results when 100 data points are used.


```
100  REM  FOURIER SERIES PROGRAM (FSERIES)
101  HOME : VTAB (7): PRINT  TAB( 13): INVERSE : PRINT "FOURIER SERIES": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II: HOME
106  PRINT : PRINT "FOURIER SERIES TO BE COMPUTED FROM:"
107  PRINT : PRINT "     1. DISCRETE DATA POINTS"
108  PRINT : PRINT "     2. AN EQUATION"
109  PRINT : PRINT : PRINT "EQUATION MUST BE DEFINED IN LINE 168": PRINT
     : PRINT "FOR EXAMPLE:"
110  PRINT : PRINT "168 F(Y) = 162 * COS (6.28 * I)"
111  PRINT : PRINT : INPUT "SELECT? ";Z9
112  PRINT : DIM F(100),C(100),D(100),E(100),A2(100)
113  IF Z9 <  > 1 THEN  PRINT : PRINT "COMPUTING ...": GOTO 167
114  HOME : PRINT : INPUT "NUMBER OF DATA POINTS? ";N
115  IF (N / 2) =  INT ((N / 2) + .5) THEN 114
116  PRINT : INPUT "NUMBER OF HARMONICS? ";N8: PRINT
117  IF N8 > (N / 2) - 2 THEN 116
118  IF Z9 <  > 1 THEN 123
119  FOR I = 1 TO N
120  PRINT "POINT #";I;"?";
121  INPUT " ";F(I)
122  NEXT I
123  PRINT : INPUT "SIGNAL PERIOD? ";T
124  N3 = N - 1
125  A = 2 * 3.1415927 / T
126  T3 = T / N3
127  A0 = 0
128  FOR I = 1 TO N3 STEP 2
129  A0 = A0 + F(I) + 4 * F(I + 1) + F(I + 2)
130  NEXT I
131  A0 = A0 * T3 / (3 * T)
132  IF  ABS (A0) < .001 THEN A0 = 0
133  HOME : VTAB (9): PRINT  TAB( 8): FLASH : PRINT "COMPUTATIONS IN PROG
     RESS": NORMAL
134  PRINT : FOR X = 1 TO N8
135  HTAB 19: PRINT X;
136  C = 0:D = 0:E = 0:A2 = 0:V = X * A
137  T1 =  - T3
138  FOR J = 1 TO N3 STEP 2
139  T1 = T1 + 2 * T3
140  C = C + (F(J) *  COS (V * (T1 - T3))) + (4 * F(J + 1) *  COS (V * T1)
     ) + (F(J + 2) *  COS (V * (T1 + T3)))
141  D = D + (F(J) *  SIN (V * (T1 - T3))) + (4 * F(J + 1) *  SIN (V * T1)
     ) + (F(J + 2) *  SIN (V * (T1 + T3)))
142  NEXT J
143  C = 2 * C * T3 / (3 * T):D = 2 * D * T3 / (3 * T)
144  IF  ABS (C) < .005 THEN C = 0
145  IF  ABS (D) < .005 THEN D = 0
146  E =  SQR (C * C + D * D)
147  IF C > 0 THEN 155
148  IF C < 0 AND D < 0 THEN 156
149  IF C = 0 AND D > 0 THEN 154
150  IF C = 0 AND D = 0 THEN 151: GOTO 152
151  A2 = 0: GOTO 158
152  IF C = 0 AND D < 0 THEN 153: GOTO 157
```

Continued on next page.

Fig. 3-9. Listing for FSERIES program.

```
153  A2 = - 90: GOTO 158
154  A2 = 90: GOTO 158
155  A2 = ( ATN (D / C)) * 180 / 3.14159: GOTO 158
156  A2 = - 180 + ( ATN (D / C)) * 180 / 3.14159: GOTO 158
157  A2 = 180 + ( ATN (D / C)) * 180 / 3.14159
158  C(X) = C:D(X) = D:E(X) = E:A2(X) = A2
159  D(X) = INT (D(X) * 1E4 + .5) / 1E4:E(X) = INT (E(X) * 1E4 + .5) / 1
     E4:C(X) = INT (C(X) * 1E4 + .5) / 1E4:A2(X) = INT (A2(X) * 100 + .
     5) / 100: NEXT X
160  HOME : PRINT : PRINT "DC TERM, A(0) = ";AO
161  PRINT : PRINT TAB( 11)"COS"; TAB( 17)"SIN": PRINT "HARMONIC TERM
     TERM MAGNITUDE PHASE"
162  FOR I = 1 TO N8
163  PRINT TAB( 4)I; TAB( 11)C(I); TAB( 17)D(I); TAB( 26)E(I); TAB( 34)A
     2(I): NEXT I
164  F = 1 / T:F = INT (F * 1E4 + .5) / 1E4
165  PRINT : PRINT "FUNDAMENTAL FREQUENCY = ";F;" HZ"
166  PRINT : END : REM
167  Y = 1: FOR I = 0 TO 1 STEP .025: REM SAMPLE FUNCTION
168  F(Y) = 162 * COS (6.283185 * I): REM SAMPLE FUNCTION
169  Y = Y + 1: NEXT I:N = Y - 1: REM SAMPLE FUNCTION
170  CALL - 998: CALL - 998: GOTO 116: REM CURSOR UP & RETURN
```

Fig. 3-9 (cont). Listing for FSERIES program.

point is the peak value of +2.5 volts. Using classical methods, the sawtooth has an average value that is one-half its peak, or 1.25 volts, while the rms value is 0.577 times the peak, or 1.443 volts.

Like the square wave, we can repeat this example by defining the waveform mathematically as a straight line with a slope equal to (2.5/0.75). Therefore, lines 123–125 would read as:

```
123 FOR I=0 TO C
124 C(I)=2.5*I/C
125 NEXT I
```

By dividing the waveform into 100 increments, the output results of Fig. 3-8 show that there is no difference!

FOURIER SERIES

Any periodic waveform can be expressed in terms of a Fourier series provided that it has a finite average value over one complete cycle; there are a finite number of discontinuities, if any exist, over one complete cycle; and it has a finite number of maximum positive and negative values. When these three conditions are met, a Fourier series for a periodic waveform f(t) exists, and can be written by the following trigonometric form:

$$f(t) = A_0 + A_1\cos(\omega t) + A_2\cos(2\omega t) + \ldots + B_1\sin(\omega t) + B_2\sin(2\omega t) + \ldots \quad \text{(Eq. 3-4)}$$

consisting of a dc, or constant term, and an infinite series of harmonically related sines and cosines.

However, Equation 3-4 may be stated in a form using only cosine and phase angle terms, so that:

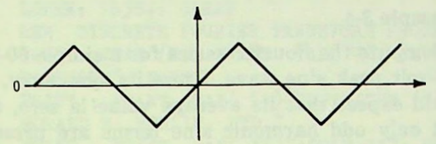
$$f(t) = C_0 + C_1\cos(\omega t + \phi_1) + C_2\cos(2\omega t + \phi_2) + \ldots \quad \text{(Eq. 3-5)}$$

Equation 3-5 is sometimes referred to as the magnitude and phase Fourier series, and is related to Equation 3-4 by:

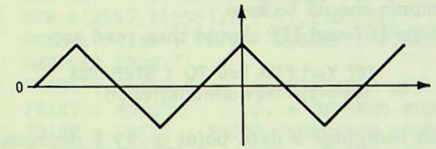
$$A_n = C_n\cos(n\omega t) \quad \text{(Eq. 3-6A)}$$
$$B_n = C_n\sin(n\omega t) \quad \text{(Eq. 3-6B)}$$
$$C_n = (A_n^2 + B_n^2)^{1/2} \quad \text{(Eq. 3-6C)}$$
$$\phi_n = \tan^{-1}(B_n/A_n) \quad \text{(Eq. 3-6D)}$$

The FSERIES program listed in Fig. 3-9 calculates the A, B, and C coefficients in addition to the phase angle ϕ for a given periodic waveform. As was the case with the RMSAV program, the waveform may be expressed either as a number of equally spaced data points, or as a periodic function of time, which must be supplied in line 168, which assumes the form that 1 complete cycle equals 6.283185 radians. Line 167 sets the sampling rate as a fraction of 1 cycle. For the values shown, the function in line 168 will be computed every 1/40th of a cycle (i.e., 1/.025), or every 9°.

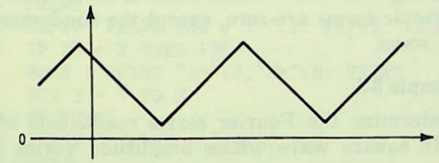
If equally spaced discrete data points are used to describe the waveform, the number of data points must be an odd number, while the number of coefficients must be less than half the number of data points minus 2. Like the RMSAV program, careful attention must be paid to functions that have discontinuities, whose number must be finite.



(A) Odd function with zero average value.



(B) Even function with zero average value.



(C) Nonzero average value.

Fig. 3-10. Symmetry functions.

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DISCRETE DATA POINTS

     2. AN EQUATION

EQUATION MUST BE DEFINED IN LINE 168
FOR EXAMPLE:

168 F(Y) = 162 * SIN (6.28 * I)

SELECT? 2

NUMBER OF HARMONICS? 7

SIGNAL PERIOD? 0.01667


DC TERM, A(0) = 0

            COS     SIN
HARMONIC    TERM    TERM    MAGNITUDE  PHASE
   1        0       162       162       0
   2        0       0         0         0
   3        0       0         0         0
   4        0       0         0         0
   5        0       0         0         0
   6        0       0         0         0
   7        0       0         0         0

FUNDAMENTAL FREQUENCY = 59.988 HZ
```

Fig. 3-11. Fourier series computed.

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DISCRETE DATA POINTS

     2. AN EQUATION

EQUATION MUST BE DEFINED IN LINE 168
FOR EXAMPLE:

168 F(Y) = 162 * SIN (6.28 * I)

SELECT? 2

NUMBER OF DATA POINTS? 21

NUMBER OF HARMONICS? 7

POINT #1? 1
POINT #2? 1
POINT #3? 1
POINT #4? 1
POINT #5? 1
POINT #6? 1
POINT #7? 1
POINT #8? 1
POINT #9? 1
POINT #10? 1
POINT #11? 0
POINT #12? -1
POINT #13? -1
POINT #14? -1
POINT #15? -1
POINT #16? -1
POINT #17? -1
POINT #18? -1
POINT #19? -1
POINT #20? -1
POINT #21? -1

SIGNAL PERIOD? 0.1

DC TERM, A(0) = 0

            COS     SIN
HARMONIC    TERM    TERM    MAGNITUDE  PHASE
   1        0       1.2733    1.2733    90
   2        0       0         0         0
   3        0       .4265     .4265     90
   4        0       0         0         0
   5        0       .2667     .2667     90
   6        0       0         0         0
   7        0       .2327     .2327     90

FUNDAMENTAL FREQUENCY = 10 HZ
```

Fig. 3-12. Results when 21 data points are used.

When the coefficients are evaluated, there may be some slight errors, especially since all computations are done with single precision arithmetic. However, by recognizing the symmetrical behavior of the particular function, one can predict which coefficients can be ignored.

Waveforms with certain types of symmetry are void of particular harmonics, such that:

1. Odd functions have only sine terms, so that all cosine coefficients are zero.
2. Even functions have only cosine terms, so that all sine terms are zero.
3. Functions that exhibit half-wave symmetry have only odd harmonics present.
4. Functions that are symmetrical about the independent variable axis have an average value (i.e., the dc or constant term) equal to zero.

Using triangle waves, Fig. 3-10 illustrates examples of those which are odd or even functions, those having only odd harmonics, and those having zero average value.

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DISCRETE DATA POINTS

     2. AN EQUATION

EQUATION MUST BE DEFINED IN LINE 168
FOR EXAMPLE:

168 F(Y) = 162 * SIN (6.28 * I)

SELECT? 2

NUMBER OF HARMONICS? 7

SIGNAL PERIOD? 0.75


DC TERM, A(0) = 1.25

          COS     SIN
HARMONIC  TERM    TERM    MAGNITUDE  PHASE
   1       0     -.7958    .7958     -90
   2       0     -.3980    .3980     -90
   3       0     -.2655    .2655     -90
   4       0     -.1995    .1995     -90
   5       0     -.1604    .1604     -90
   6       0     -.1349    .1349     -90
   7       0     -.1176    .1176     -90

FUNDAMENTAL FREQUENCY = 1.3333 HZ
```

Fig. 3-13. Results with all cosine terms zero.

Example 3-4

Compute the Fourier series for a simple 60-Hz, 162-volt peak sine wave. From its symmetry, we would expect that its average value is zero, and that only odd harmonic sine terms are present. Furthermore, since we have a pure sine wave, all harmonics except the fundamental, or first harmonic should be zero.

Lines 167 and 168 should then read as:

```
167 Y=1:FOR I=0 TO 1 STEP .025
168 F(Y)=162*SIN(6.283185*I)
```

which computes a data point every 9 degrees, or 1/40th of a cycle. Fig. 3-11 shows the results of the program. The average value is zero, while all harmonic terms are zero, except the fundamental sine wave.

Example 3-5

Determine the Fourier series coefficients of a 10-Hz square wave whose amplitude varies between −1 and +1 volts. Since such a wave is both an odd function and has half-wave symmetry, we would then expect that the average value, all cosine terms, and all even sine terms, will be zero. Fig. 3-12 shows the output results when 21 data points are used, so that:

Points 1–10: +1
Point 11: 0 (the half-cycle discontinuity)
Points 12–21: −1

As a comparison, it can be shown that the Fourier series for this square wave is:

$$f(t) = \frac{4V}{\pi} \sum_{n=0}^{m} \frac{\sin(2n+1)\omega t}{2n+1} \quad \text{(Eq. 3-7)}$$

If desired, the square wave can be expressed mathematically as was the case for determining the average and rms values in Example 3-2, and averaging the results of two runs. This is left as an optional exercise for the reader.

Example 3-6

Determine the Fourier coefficients for the sawtooth wave previously shown in Fig. 3-6. From its symmetry, we would expect that, since it is an odd function, only sine terms will be present.

Fig. 3-13 shows the results, in which all cosine terms are zero. As a comparison, it can be shown that the Fourier series for this sawtooth is:

$$f(t) = \frac{V}{2} - \frac{V}{\pi} \sum_{n=1}^{m} \frac{\sin(n\omega t)}{n} \quad \text{(Eq. 3-8)}$$


```
100  LOMEM: 16384: CLEAR
105  REM  DISCRETE FOURIER TRANSFORM PROGRAM (DFTRANS)
110  HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "DISCRETE FOURIER T
     RANSFORM": NORMAL
115  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
120  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 12)"APPLESOFT
     VERSION"
125  FOR II = 1 TO 2500: NEXT II
130  DIM A(256),B(256),X(256),Y(256):T = 1000:V = 8
135  HOME : PRINT : PRINT "DISCRETE FOURIER TRANSFORM": PRINT "IS TO BE C
     OMPUTED FROM:"
140  PRINT : PRINT "     1. DISCRETE DATA POINTS"
145  PRINT : PRINT "     2. A DEFINED FUNCTION"
150  PRINT : PRINT : PRINT "FUNCTION MUST BE DEFINED": PRINT "ACCORDING T
     O INSTRUCTIONS ": FLASH : PRINT "IN PAMPHLET": NORMAL
155  PRINT : INPUT "SELECT? ";Z9
160  PRINT
165  INPUT "VALUE FOR J ? ";J: INPUT "VALUE FOR N ? ";N
170  IF Z9 = 2 THEN 195
172  HOME : PRINT "J=";J,"N=";N: PRINT
175  FOR Z = 1 TO N
180  PRINT Z;
185  INPUT " VALUE = ";A(Z)
190  NEXT Z
195  P = 3.1415926535
200  HOME : VTAB (7): PRINT  TAB( 5)"CALCULATION OF INPUT FUNCTION": PRINT
     : PRINT  TAB( 14): FLASH : PRINT "IN PROGRESS": NORMAL
205  IF Z9 = 1 THEN 240
210  FOR Z = 0 TO N * 7 / 8
215  A(Z) = 1
220  NEXT Z
225  FOR Z = N * 7 / 8 TO N
230  A(Z) = 0
235  NEXT Z
240  FOR Z = 0 TO N - 1
245  A(Z) = A(Z) / N
250  NEXT Z
255  C = N / 2:D = 1:E = P * 2 / N: PRINT
260  FOR I = 1 TO J: HTAB (19): PRINT I: CALL  - 998
265  F = 0:G = C
270  FOR S = 1 TO D
275  H =  INT (F / C)
280  GOSUB 465
285  Q = R
290  A1 =  COS (E * Q):A2 =  -  SIN (E * Q)
295  FOR K = F TO G - 1
300  A3 = A(K):A4 = B(K)
305  B1 = A1 * A(K + C) - A2 * B(K + C)
310  B2 = A2 * A(K + C) + A1 * B(K + C)
315  A(K) = A3 + B1:B(K) = A4 + B2
320  A(K + C) = A3 - B1:B(K + C) = A4 - B2
325  NEXT K
330  F = F + 2 * C:G = G + 2 * C
335  NEXT S
340  C = C / 2:D = D * 2
345  NEXT I
350  GOTO 355
355  U = 0
```

Continued on next page.

Fig. 3-14. Listing for DFTRANS program.

```
360  Z = 0
365  HOME
370  PRINT "HM"; TAB( 5)"RE(Z)"; TAB( 19)"IM(Z)"; TAB( 33)"Z"
375  PRINT "=="; TAB( 5)"=====";  TAB( 19)"=====";  TAB( 32)"====="
380  H = U
385  GOSUB 505
390  V1$ =  LEFT$ (( STR$ (U)),V)
391  V2 = ( INT ((A(R)) * 10 ^ 7 + .5)) / 10 ^ 7
392  V3 = ( INT ((B(R)) * 10 ^ 7 + .5)) / 10 ^ 7
393  V4 = ( INT (ZZ * 10 ^ 7 + .5)) / 10 ^ 7
395  PRINT  TAB( 1)V1$; TAB( 4)V2; TAB( 16)V3; TAB( 28)V4
400  Y(U) = ZZ:X(U) = U
405  U = U + 1:Z = Z + 1
410  IF Z <  = 13 THEN 430
415  PRINT : PRINT "HIT RETURN TO CONTINUE";:
420  POKE  - 16368,0: CALL  - 756: HOME
425  GOTO 360
430  IF U > N / 2 THEN 440
435  GOTO 380
440  PRINT : PRINT "HIT RETURN FOR SPECTRUM PLOT"
445  POKE  - 16368,0:N = U: GOTO 530
455  GOTO 530
465  R = 0:N1 = N
470  FOR W = 1 TO J
475  N1 = N1 / 2
480  IF H < N1 THEN 495
485  R = R + 2 ^ (W - 1)
490  H = H - N1
495  NEXT W
500  RETURN
505  GOSUB 465
510  ZZ =  SQR (A(R) ^ 2 + B(R) ^ 2)
515  RETURN
520  END
525  REM    SPECTRUM PLOT ROUTINE
530  Y1 = Y(0):Y2 = Y(0):X1 = X(0):X2 = X(0)
535  FOR I = 1 TO N
540  IF (Y1 - Y(I)) <  = 0 THEN 555
545  Y1 = Y(I)
550  GOTO 565
555  IF (Y2 - Y(I)) >  = 0 THEN 565
560  Y2 = Y(I)
565  IF (X(I)) <  = 0 THEN 580
570  X1 = X(I)
575  GOTO 590
580  IF (X2 - X(I)) >  = 0 THEN 590
585  X2 = X(I) + 1
590  : NEXT I: CALL  - 756
595  XD = X1 - X2:YD = Y2 - Y1
600  CALL 62450: HGR : HCOLOR= 7
605  HPLOT 0,145 TO 278,145
610  FOR X = 0 TO 278 STEP 278 / 8: HPLOT X,150 TO X,145: NEXT X
615  FOR I = 0 TO N - 1
620  X = 278 - ((X1 - X(I)) * 278 / XD):Y = 141 - ((Y(I) - Y1) * 141 / YD)
625  IF X > 279 THEN 635
630  HPLOT X,Y TO X,141
635  NEXT I
```

Continued on next page.

Fig. 3-14 (cont). Listing for DFTRANS program.


```
640  TB = 39: IF X1 > 99 THEN TB = 38
645  VTAB (21): PRINT "0"; TAB( 14)"HARMONIC NUMBER"; TAB( TB)X1
650  POKE  - 16368,0: CALL  - 756
655  HOME : TEXT : VTAB (12): PRINT "GOODBYE ........"
660  END
```

Fig. 3-14 (cont). Listing for DFTRANS program.

FOURIER TRANSFORM

Whereas the Fourier series is used to represent the harmonic, or spectral content, of any periodic function, the Fourier transform is intended for those signals which are not periodic, but nevertheless continuous. The mathematical transformation of a function of time f(t) into a function of frequency F(s) is given by the equation:

$$F(s) = \int_{-\infty}^{\infty} f(t)e^{-st}\,dt \qquad \text{(Eq. 3-9)}$$

Instead of being able to represent the Fourier transform of a continuous function as a continuous function of frequency, the DFTRANS program of Fig. 3-14 computes a discrete Fourier

HARM	RE(Z)	IM(Z)	Z
0	.875	0	.875
1	-.11016	-.0521	.12186
2	-.07151	-.07713	.11272
3	-.0239	-.09543	.09839
4	.01562	-.07855	.08009
5	.03539	-.04772	.05941
6	.03357	-.01794	.03806
7	.1373	-8.7E-04	.01375
8	0	0	0
9	-9.39E-03	-.01036	.01399

Fig. 3-15. Output results for first nine harmonics.

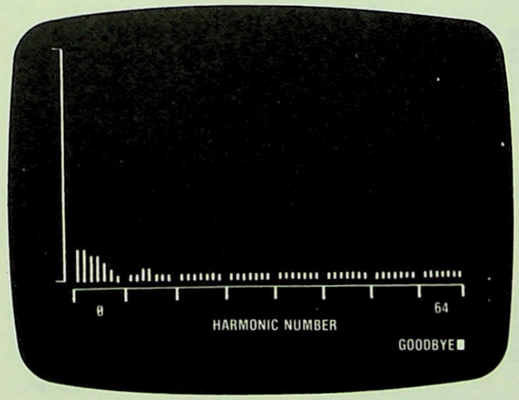


Fig. 3-16. Spectrum plot.

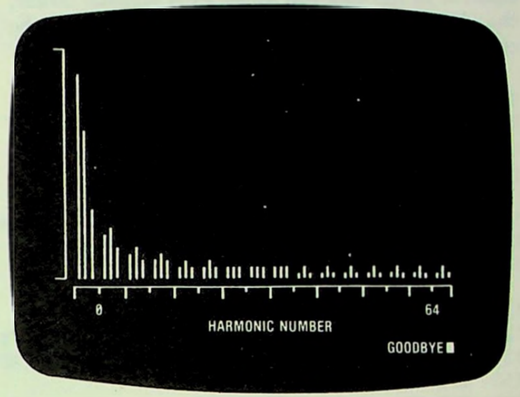


Fig. 3-17. Spectrum plot for a 25% duty cycle pulse.

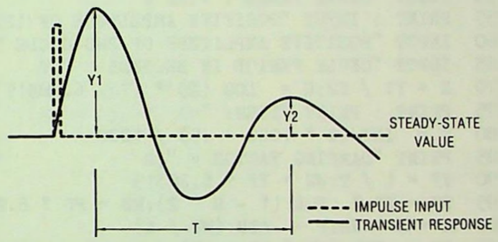


Fig. 3-18. Transient response to an impulse input.

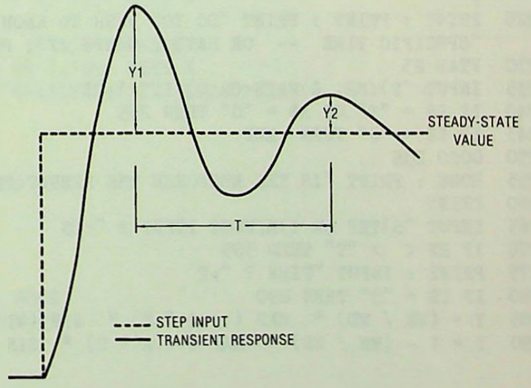


Fig. 3-19. Transient response to a step input.

transform so that the frequency function is now approximated by discrete points instead of a continuous function.

The continuous time function is sampled N times, so that $N = 2^J$, where J is an integer. If the number of sample points is 64, then J must equal 6. In the classical sense, as N is made larger, the discrete Fourier transform approaches the continuous function described by Equation 3-9.

When run, the DFTRANS program requires that the number of data point samples (N) in line 165 be as shown in Table 3-1; consequently, the value for J must also be defined as summarized in Table 3-1. Because this program generates a spectrum plot that can contain up to 128 discrete frequencies, the maximum number of sampled data points is 256, in which case J = 8.

Table 3-1. DFTRANS Program

Data Points		Discrete Frequencies
N	J	
16	4	8
32	5	16
64	6	32
128	7	64
256	8	128

```
100  REM  ANALYSIS OF DAMPED OSCILLATIONS (DAMPED)
105  LOMEM: 16385
110  HOME : VTAB 6: PRINT  TAB( 6): INVERSE : PRINT "ANALYSIS OF DAMPED O
     SCILLATIONS": NORMAL
115  VTAB 20
120  PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
125  VTAB (13): PRINT  SPC( 12)"APPLESOFT VERSION"
130  DIM X(267),Y(267)
135  FOR II = 1 TO 2500: NEXT
140  HOME
145  PRINT "AMPLITUDES ARE MEASURED FROM THE STEADY"
150  PRINT "STATE VALUE": VTAB 6
155  PRINT : INPUT "POSITIVE AMPLITUDE OF 1ST CYCLE ? ";Y1
160  INPUT "POSITIVE AMPLITUDE OF 2ND CYCLE ? ";Y2
165  INPUT "CYCLE PERIOD IN SECONDS ? ";T
170  X = Y1 / Y2:G =  LOG (X) * (1 / 6.28319):D = G /  SQR (1 + G ^ 2)
175  PRINT : PRINT "THEN: "
180  G =  INT (G * 1000 + .5) / 1000
185  PRINT "DAMPING FACTOR = ";G
190  FF = 1 / T:WN = FF * 6.28319
195  FD = FF *  SQR (1 - D ^ 2):WD = FD * 6.28319
200  A = D * WN:P =  ATN (WN / A)
205  FF =  INT (FF * 10 + .5) / 10:WN =  INT (WN * 10 + .5) / 10
210  FD =  INT (FD * 10 + .5) / 10:WD =  INT (WD * 10 + .5) / 10
215  PRINT "NATURAL FREQUENCY = ";FF;" HZ": PRINT "   (";WN;" RAD/SEC)"
220  PRINT "DAMPED NATURAL FREQ  = ";FD;" HZ ": PRINT "   (";WD;" RAD/SEC
     )"
225  PRINT : PRINT : PRINT "DO YOU WISH TO KNOW THE RESPONSE AT A": PRINT
     "SPECIFIC TIME  --  OR HAVE A GRAPH OF": PRINT "THE WAVEFORM  ?"
230  VTAB 23
235  INPUT "T)IME, G)RAPH OR Q)UIT? ";R$
240  IF R$ = "T" OR R$ = "G" THEN 255
245  IF R$ = "Q" THEN  END
250  GOTO 235
255  HOME : PRINT "IS THE RESPONSE THE RESULT OF A:"
260  PRINT
265  INPUT "S)TEP OR I)MPULSE INPUT ? ";I$
270  IF R$ <  > "T" THEN 305
275  PRINT : INPUT "TIME ? ";T
280  IF I$ = "S" THEN 290
285  Y = (WN / WD) *  EXP ( - A * T) *  SIN (WD * T): GOTO 295
290  Y = 1 - (WN / WD) *  EXP ( - A * T) *  SIN (WD * T + P)
```

Continued on next page.

Fig. 3-20. Listing for DAMPED program.


```
295  PRINT "AMPLITUDE = ";Y
300  GOTO 225
305  HOME : INPUT "ENTER MIN,MAX TIME ";XA,XB
310  PRINT
315  N = 266:I = 1:C = (XB - XA) / 265
320  FOR XT = XA TO XB + C STEP C
325  IF I$ = "S" THEN 335
330  Y = (WN / WD) *  EXP ( - A * XT) *  SIN (WD * XT): GOTO 340
335  Y = 1 - (WN / WD) *  EXP ( - A * XT) *  SIN (WD * XT + P)
340  X(I) = XT:Y(I) = Y:I = I + 1: PRINT XT;"         ": CALL  - 998: NEXT
     XT
345  CALL  - 868: HTAB 1
350  GOSUB 365
355  PRINT : END
360  REM  X-Y PLOT SUBROUTINE
365  T$ = "STEP RESPONSE"
370  IF I$ = "I" THEN T$ = "IMPULSE RESPONSE"
375  REM  FIND MIN,MAX OF X & Y
380  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
385  FOR I = 2 TO N
390  IF (Y1 - Y(I)) <  = 0 THEN 405
395  Y1 = Y(I)
400  GOTO 415
405  IF (Y2 - Y(I)) >  = 0 THEN 415
410  Y2 = Y(I)
415  IF (X1 - X(I)) <  = 0 THEN 430
420  X1 = X(I)
425  GOTO 440
430  IF (X2 - X(I)) >  = 0 THEN 440
435  X2 = X(I): PRINT I: CALL  - 998
440  NEXT I
445  X1 =  INT (X1 * 1E6 + .5) / 1E6:X2 =  INT (X2 * 1E6 + .5) / 1E6
450  Y1 =  INT (Y1 * 1E6 + .5) / 1E6:Y2 =  INT (Y2 * 1E6 + .5) / 1E6
455  PRINT "MIN-MAX X VALUES: ": PRINT X1,X2: PRINT
460  PRINT "MIN-MAX Y VALUES: ": PRINT Y1,Y2: PRINT
465  INPUT "DESIRED MIN.MAX X-AXIS ? ";X1,X2
470  INPUT "DESIRED MIN,MAX Y-AXIS ? ";Y1,Y2
475  XD = X2 - X1:YD = Y2 - Y1
480  REM  ------ GRAPH ---------
485  GR : HGR : HCOLOR= 7: VTAB (23)
490  HPLOT 8,0 TO 8,150
495  FOR Y = 150 TO 0 STEP  - (150 / 10): HPLOT 6,Y TO 7,Y: NEXT Y
500  FOR Y = 150 TO 0 STEP  - (150 / 5): HPLOT 3,Y TO 7,Y: NEXT Y
505  HPLOT 13,154 TO 279,154
510  FOR X = 13 TO 279 STEP 266 / 10: HPLOT X,155 TO X,157: NEXT X
515  FOR X = 13 TO 279 STEP 266 / 5: HPLOT X,155 TO X,159: NEXT X
520  VTAB 24: PRINT "Y-AXIS ";Y1;":";Y2; SPC( 5)"X-AXIS ";X1;":";X2: PRINT
525  FOR I = 1 TO N
530  X = 279 - ((X2 - X(I)) * 266 / XD)
535  Y = 150 - ((Y(I) - Y1) * 150 / YD)
540  IF X > 279 OR X < 13 OR Y < 0 OR Y > 159 THEN 552
545  IF I = 1 THEN  HPLOT X,Y: GOTO 552
550  HPLOT  TO X,Y
552  NEXT I
555  INVERSE : PRINT T$;: NORMAL
560  PRINT "   CONTINUE ?";: CALL  - 756: TEXT : HOME
565  INPUT "ANY CHANGES IN SCALE FACTORS (Y/N) ? ";A$
570  IF A$ = "Y" THEN 455
575  RETURN
```

Fig. 3-20 (cont). Listing for DAMPED program.

We have the choice to compute the Fourier transform from either by entering discrete data points describing the waveform, or mathematically defining the function. Lines 210 to 235 describe, as an example, a rectangular pulse having a duty cycle of 87.5%, so that the amplitude is 1 for 7/8 of the period and zero for the remainder. After the input function is processed, the Fourier transform is tabulated in terms of its real and imaginary components, as well as its magnitude for each harmonic starting with the fundamental (i.e., harmonic "0"). In addition, these values are displayed in groups of 20. Pressing the RETURN key continues the listing until N/2 harmonics are displayed. At the end of the listing, hitting the RETURN key once more gives the magnitude spectrum plot as a function of its harmonic number.

Example 3-7

Compute the discrete Fourier transform for the function already described in Fig. 3-14, up to and including the 64th harmonic.

Fig. 3-15 shows the output results for the first 9 harmonics plus the fundamental while the spectrum plot is shown in Fig. 3-16. Fig. 3-17 shows the spectrum plot for a 25% duty cycle pulse (64 harmonics).

ANALYSIS OF DAMPED OSCILLATIONS

When an underdamped second-order system encounters a sudden change at its input, such as the voltage controlled oscillator of a phase-locked loop shifting from one frequency to another, the output tries to follow this change, but oscillates about a steady-state value for a time and eventually settles out. This process is illustrated in Fig. 3-18 for an impulse response, and in Fig. 3-19 for a step response. How fast this process is completed depends on the damping factor of the system. For smaller values of damping, it takes longer for the oscillations to settle down.

The DAMPED program listing in Fig. 3-20 determines the damping factor, the natural and damped frequencies. In addition, the program determines the normalized amplitude of the waveform at a given time. The required inputs are the positive amplitudes of two consecutive peaks, which will be one cycle apart, and the time period for one cycle. If desired, the resulting damped oscillation is plotted on the video display.

Example 3-8

From oscilloscope measurements of a damped oscillation resulting from a step response, it was measured that two consecutive peaks, like those of Fig. 3-19, had amplitudes of 8 and 3 divisions, respectively. In addition, the period for one complete cycle was measured as 0.46 ms. We want to know what will the amplitude be after 1.5 ms, when the peak input is 0.253 volt. Fig. 3-21 shows the results of the analysis.

```
AMPLITUDES ARE MEASURED FROM THE STEADY
STATE VALUE


POSITIVE AMPLITUDE OF 1ST CYCLE ? 8
POSITIVE AMPLITUDE OF 2ND CYCLE ? 3
CYCLE PERIOD IN SECONDS ? 0.46E-3

THEN:
DAMPING FACTOR = .156
NATURAL FREQUENCY = 2173.9 HZ
   (13659.1 RAD/SEC)
DAMPED NATURAL FREQ = 2147.9 HZ
   (13495.7 RAD/SEC)

DO YOU WISH TO KNOW THE RESPONSE AT A
SPECIFIC TIME  --  OR HAVE A GRAPH OF
THE WAVEFORM  ?

T)IME, G)RAPH OR Q)UIT? T
IS THE RESPONSE THE RESULT OF A:

S)TEP OR I)MPULSE INPUT ?I

TIME ? 1.5E-3
AMPLITUDE = 0.422664465

DO YOU WISH TO KNOW THE RESPONSE AT A
SPECIFIC TIME  --  OR HAVE A GRAPH OF
THE WAVEFORM  ?

T)IME, G)RAPH OR Q)UIT? G
IS THE RESPONSE THE RESULT OF A:

S)TEP OR I)MPULSE ? I

ENTER MIN,MAX TIME ?0,2E-3

MIN-MAX X VALUES:
0              2E-03
MIN-MAX Y VALUES:
-.490691       .799867

DESIRED MIN,MAX X-AXIS ?0,2E-3
DESIRED MIN,MAX Y-AXIS ?-.5,1
```

Fig. 3-21. Analysis of DAMPED oscillations.

4

Basic Statistics and the Analysis of Data

In any experimental endeavor, there usually comes a time when we must ask ourselves whether or not the data is a true representation of the experiment. Although this chapter is not meant to be a course in statistics, it is nevertheless important to include several useful statistical methods. Included are programs for computing the mean and standard deviation of a set of data, and regression analysis.

LEAST SQUARES REGRESSION CURVES

Almost every experimenter seeks to establish a relationship between two or more variables, such as the variation of impedance with frequency, or the variation of a transistor operating point as a function of ambient temperature and its current gain. As an aid in establishing a relationship (or lack of it), it is often desirable to express the relationship in the form of an equation.

Relationship Between Two Variables

Listed in Chart 4-1 are several common forms of mathematical relationships used to relate the behavior of two variables. To decide which equation should be used, it is helpful to graph the data in the form of a scatter diagram, as shown in Fig. 4-1, that can be compared against one of the general curves shown in Fig. 4-2.

Fig. 4-3 lists the REGRESSN program for determining the least squares regression of any one

Chart 4-1. Common Forms of Mathematical Relationships

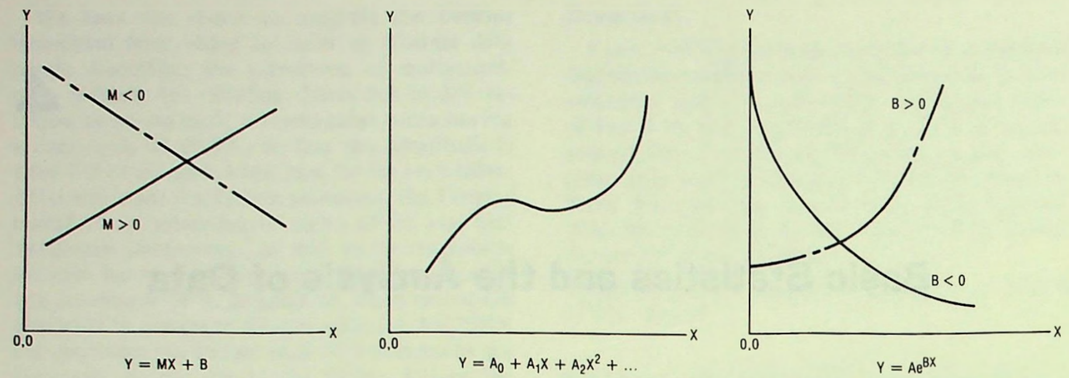
Linear	$y = mx + b$
Inverse	$y = 1/(mx + b)$
Nth order (polynomial)	$y = a_0 + a_1x + a_2x^2 + \ldots$
Geometric	$y = ax^b$
Exponential	$y = ae^{bx}$

Fig. 4-1. Scatter diagram.

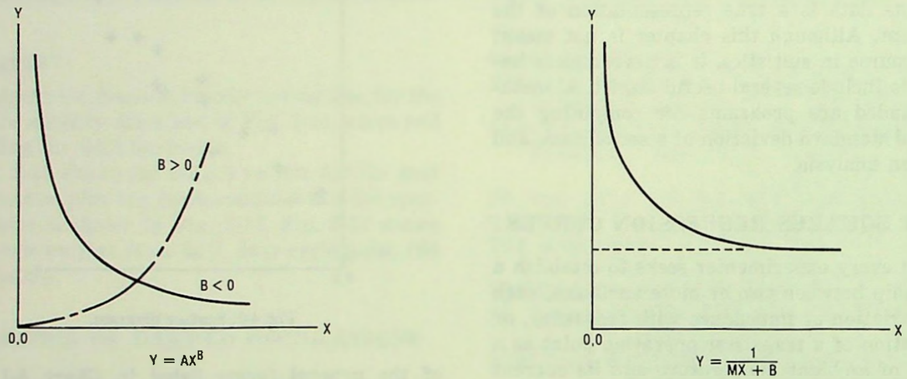
of the general forms listed in Chart 4-1. The REGRESSN program does the following:

1. Stores and retrieves data from disk.
2. Adds, deletes, or changes data.
3. Determines the minimum and maximum values of data points.
4. Plots a scatter diagram of the entered data.
5. Determines the number of data points entered.
6. Computes the coefficient of determination, correlation coefficient, and standard error of estimate.
7. Interpolates.

The REGRESSN program, like the plotting programs of Chapter 2, uses HiRes graphics for plotting the scatter diagram. The program is loaded at the decimal memory location 2048 and is approximately 5967 bytes long. It must be remembered that the HiRes graphics begins at decimal memory location 8192, so that there are only approximately 177 free bytes for adding lines to



(*A*) *Linear.* (*B*) *Polynomial.* (*C*) *Exponential.*



(*D*) *Geometric.* (*E*) *Inverse.*

Fig. 4-2. Generalized regression curves.

the program before REGRESSN overlaps the HiRes display area.

You may want to delete the REM lines from REGRESSN to enable you to add more lines to it. Before running the modified program, you may check it by typing:

```
PRINT PEEK(105) + PEEK(106)*256
```

and then hitting the RETURN key. The number you will see is the current LOMEM. If LOMEM is greater than 8192, then you must delete some of your program and try again. Remember, LOMEM in Applesoft is the end of the program area and is pushed up or down as you add or delete lines.

Example 4-1

Determine what kind of relationship exists between the following 10 X-Y data points, which are the parameters of voltage in millivolts, and temperature in degrees Fahrenheit for a given copper-constantan thermocouple: (−2.581, −100), (−1.101, −20), (−0.674, 0), (−0.404, 13), (−0.150, 25), (−0.043, 30), (0, 31), (1.086, 80), (1.542, 101), and (3.711, 190).

When the REGRESSN program is run, we are first asked if we want to enter data that has been previously stored on disk. If the above data has not been previously stored, the typed response is the letter N, in which case we are asked to enter each point, as shown in Fig. 4-4. After all data points have been entered, hit the RETURN key to terminate. All the data points are then listed so that we may then examine them for any possible errors or omissions. For example, suppose point No. 7 was supposed to be (0, 32) instead of (0, 31). We then type the letter Y in response to whether or not we want to change a data point.


```
  0 LOMEM: 16384: REM    BEFORE   SETTING ANY VARIABLES
  1 REM  **** CAUTION ****
100 REM ADDING TO THIS PROGRAM
101 REM MAY CAUSE PROBLEMS
102 REM   FROM HIRES PAGE1
103 REM  OVERLAYING PROGRAM
104 TEXT :D$ =  CHR$ (13) +  CHR$ (4)
105 HOME : VTAB 6: PRINT  TAB( 6): INVERSE : PRINT "LINEAR & CURVILINEAR
    REGRESSION": NORMAL
106 VTAB 20:Z9 = 1
107 PRINT "COPYRIGHT (C) 1980 HOWARD M.BERLIN, LTD.
108 VTAB (13): PRINT  SPC( 11)"APPLESOFT VERSION"
109 FOR II = 1 TO 2500: NEXT
110 DIM X(100),Y(100),P(100,2),V1(100),A(21),U(11,12),T(12)
111 AI$ = "ANOTHER INTERPOLATION (Y/N) "
112 GOSUB 290: GOSUB 224: GOSUB 241
113 GOTO 257
114 HOME : PRINT "LINEAR REGRESSION"
115 GOSUB 315
116 FOR I = 1 TO N:X = X(I):Y = Y(I):J = J + X:K = K + Y:L = L + X ^ 2:M
    = M + Y ^ 2:R2 = R2 + X * Y: NEXT I
117 GOSUB 263
118 PRINT : PRINT "Y = ";A;" + (";B;" * X)"
119 GOSUB 265
120 INPUT "INTERPOLATION (Y/N) ";C$
121 IF C$ <  > "Y" THEN 126
122 INPUT "X= ";X
123 PRINT "     THEN Y = ";A + B * X
124 PRINT AI$;: INPUT C$
125 IF C$ = "Y" THEN 122
126 INPUT "SAVE INPUT DATA TO DISK (Y/N) ";ZZ$
127 IF ZZ$ = "Y" THEN  GOSUB 274
128 GOTO 262
129 GOSUB 315
130 FOR I = 1 TO N:X = X(I):Y = Y(I):Y =  LOG (Y):X =  LOG (X):J = J + X
    :K = K + Y:L = L + X ^ 2:M = M + Y ^ 2:R2 = R2 + X * Y: NEXT I
131 GOSUB 263
132 HOME : PRINT "GEOMETRIC REGRESSION (FOR X,Y>0)": PRINT
133 PRINT "       B": PRINT "Y = AX"
134 PRINT : PRINT "     A = "; EXP (A)
135 PRINT "      B = ";B
136 GOSUB 265
137 INPUT "INTERPOLATION (Y/N) ";C$
138 IF C$ <  > "Y" THEN 143
139 INPUT "X = ";X
140 PRINT "      THEN Y = "; EXP (A) * (X ^ B)
141 PRINT AI$;: INPUT C$: IF C$ <  > "Y" THEN 143
142 GOTO 139
143 INPUT "SAVE DATA TO DISK (Y/N) ";ZZ$
144 IF ZZ$ = "Y" THEN  GOSUB 274
145 GOTO 262
146 GOSUB 315
147 FOR I = 1 TO N:X = X(I):Y = Y(I):Y =  LOG (Y):J = J + X:K = K + Y:L =
    L + X ^ 2:M = M + Y ^ 2:R2 = R2 + X * Y: NEXT I: GOSUB 263
148 HOME : PRINT "EXPONENTIAL REGRESSION (FOR Y<>0)": PRINT
149 PRINT "          BX"
150 PRINT "Y = A(EXP)"
151 PRINT : PRINT "      A ="; EXP (A): PRINT "      B =";B
152 GOSUB 265
```

Continued on next page.

Fig. 4-3. Listing for REGRESSN program.

```
153  INPUT "INTERPOLATION (Y/N) ";C$
154  IF C$ <  > "Y" THEN 159
155  INPUT "X = ";X
156  PRINT "     THEN Y = "; EXP (A) *  EXP (B * X)
157  PRINT AI$;: INPUT C$: IF C$ <  > "Y" THEN 159
158  GOTO 155
159  INPUT "SAVE DATA TO DISK (Y/N) ";ZZ$
160  IF ZZ$ = "Y" THEN  GOSUB 274
161  GOTO 262
162  HOME : PRINT "POLYNOMIAL REGRESSION"
163  PRINT  SPC( 22)"N"
164  PRINT "Y = A  + A X +.....A X"
165  PRINT "     0    1          N"
166  PRINT : INPUT "ORDER OF POLYNOMIAL ";D
167  HOME : VTAB (9): PRINT  TAB( 15): FLASH : PRINT "CALCULATING...": NORMAL
168  A(1) = N:J = 0:K = 0:Q = 0:S = 0:Z = 0:I = 0:CC = 0
169  FOR I = 1 TO N
170  X = X(I):Y = Y(I)
171  FOR J = 2 TO 2 * D + 1:A(J) = A(J) + X ^ (J - 1): NEXT J
172  FOR K = 1 TO D + 1:U(K,D + 2) = T(K) + Y * X ^ (K - 1):T(K) = T(K) +
     Y * X ^ (K - 1): NEXT K
173  T(D + 2) = T(D + 2) + Y ^ 2: NEXT I
174  FOR J = 1 TO D + 1: FOR K = 1 TO D + 1:U(J,K) = A(J + K - 1): NEXT K
     : NEXT J
175  FOR J = 1 TO D + 1: FOR K = J TO D + 1
176  IF U(K,J) <  > 0 THEN 179
177  NEXT K
178  INVERSE : PRINT "NO UNIQUE SOLUTION CAN BE FOUND": NORMAL : GOTO 206
179  FOR I = 1 TO D + 2:S = U(J,I):U(J,I) = U(K,I):U(K,I) = S: NEXT I
180  Z = 1 / U(J,J)
181  FOR I = 1 TO D + 2:U(J,I) = Z * U(J,I): NEXT I
182  FOR K = 1 TO D + 1
183  IF K = J THEN 186
184  Z =  - U(K,J)
185  FOR I = 1 TO D + 2:U(K,I) = U(K,I) + Z * U(J,I): NEXT I
186  NEXT K: NEXT J
187  HOME : PRINT "POLYNOMIAL COEFFICIENTS:": PRINT "A(0) = ";U(1,D + 2)
188  FOR J = 1 TO D: PRINT "A(";J;") = ";U(J + 1,D + 2): NEXT J
189  P = 0: FOR J = 2 TO D + 1:P = P + U(J,D + 2) * (T(J) - A(J) * T(1) /
     N): NEXT J
190  Q = T(D + 2) - T(1) ^ 2 / N:Z = Q - P:I = N - D - 1:L = P / Q
191  PRINT : INPUT "HIT RETURN TO CONTINUE ";C$
192  PRINT  SPC( 21)"2"
193  PRINT "DETERMINATION COEFF R = ";L
194  PRINT "CORRELATION COEFFICIENT R = "; SQR (L)
195  PRINT "STANDARD ESTIMATE ERROR = "; SQR (Z / I)
196  INPUT "INTERPOLATION (Y/N) ";C$
197  IF C$ <  > "Y" THEN 204
198  INPUT "X = ";X
199  P = U(1,D + 2)
200  FOR J = 1 TO D:P = P + U(J + 1,D + 2) * X ^ J: NEXT J
201  PRINT "     THEN Y = ";P
202  PRINT AI$;: INPUT C$
203  IF C$ = "Y" THEN 198
204  INPUT "SAVE INPUT DATA TO DISK (Y/N) ";ZZ$
205  IF ZZ$ = "Y" THEN  GOSUB 274
206  :
207  GOTO 262
```

Continued on next page.

Fig. 4-3 (cont). Listing for REGRESSN program.


```
208  HOME : PRINT "INVERSE REGRESSION"
209  GOSUB 315
210  FOR I = 1 TO N:X = 1 / X(I):Y = Y(I)
211  J = J + X:K = K + Y:L = L + X ^ 2:M = M + Y ^ 2:R2 = R2 + X * Y: NEXT
     I
212  GOSUB 263
213  PRINT : PRINT "Y = ";A;" + (";B;"/X)"
214  GOSUB 265
215  INPUT "INTERPOLATION (Y/N) ";C$
216  IF C$ <  > "Y" THEN 221
217  INPUT "X = ";X
218  PRINT "     THEN Y = ";A + B / X
219  PRINT AI$;: INPUT C$
220  IF C$ = "Y" THEN 217
221  INPUT "SAVE DATA TO DISK (Y/N) ";ZZ$
222  IF ZZ$ = "Y" THEN  GOSUB 274
223  GOTO 262
224  Y1 = Y(1):Y2 = Y(1):X1 = X(1):X2 = X(1)
225  FOR I = 1 TO N
226  IF (Y1 - Y(I)) <  = 0 THEN 228
227  Y1 = Y(I): GOTO 230
228  IF (Y2 - Y(I)) >  = 0 THEN 230
229  Y2 = Y(I)
230  IF (X1 - X(I)) <  = 0 THEN 233
231  X1 = X(I)
232  GOTO 235
233  IF (X2 - X(I)) >  = 0 THEN 235
234  X2 = X(I)
235  NEXT I
236  HOME : PRINT "YOUR DATA HAS THE FOLLOWING LIMITS:"
237  PRINT : PRINT : PRINT "VARIABLE","MINIMUM","MAXIMUM"
238  PRINT "X",X1,X2: PRINT "Y",Y1,Y2
239  XD = X2 - X1:YD = Y2 - Y1
240  RETURN
241  PRINT : INPUT "HIT RETURN FOR A GRAPH OF INPUT DATA";FF$
242  HGR : HCOLOR= 7: VTAB (23)
243  HPLOT 8,1 TO 8,150
244  FOR Y = 150 TO 0 STEP  - (150 / 8): HPLOT 6,Y TO 7,Y: NEXT Y
245  FOR Y = 150 TO 0 STEP  - (150 / 4): HPLOT 3,Y TO 7,Y: NEXT Y
246  HPLOT 13,154 TO 279,154
247  FOR X = 13 TO 279 STEP 266 / 8: HPLOT X,155 TO X,157: NEXT X
248  FOR X = 13 TO 279 STEP 266 / 4: HPLOT X,157 TO X,159: NEXT X
249  VTAB 24: PRINT "Y-AXIS ";Y1;":";Y2; SPC( 5)"X-AXIS ";X1;":";X2: PRINT
250  FOR I = 1 TO N
251  X = 279 - ((X2 - X(I)) * 266 / XD)
252  Y = 150 - ((Y(I) - Y1) * 150 / YD)
253  IF X > 279 OR X < 13 OR Y < 0 THEN 5290
254  YT = Y
255  HPLOT X,Y: NEXT I
256  PRINT "HIT RETURN TO CONTINUE ";: CALL  - 756: TEXT : HOME : RETURN
257  HOME : PRINT "REGRESSION TYPES AVAILABLE:"
258  PRINT : PRINT "1)  LINEAR REGRESSION": PRINT "2)  INVERSE REGRESSION
     ": PRINT "3)  NTH ORDER (POLYNOMIAL) REGRESSION": PRINT "4)  GEOMETR
     IC REGRESSION": PRINT "5)  EXPONENTIAL REGRESSION": PRINT : PRINT
259  INPUT "YOUR CHOICE? ";ZX
260  IF ZX < 1 OR ZX > 5 THEN 257
261  ON ZX GOTO 114,208,162,129,146
```

Continued on next page.

Fig. 4-3 (cont). Listing for REGRESSN program.

```
262  PRINT : END
263  B = (N * R2 - K * J) / (N * L - J ^ 2)
264  A = (K - B * J) / N: RETURN
265  J = B * (R2 - J * K / N)
266  M = M - K ^ 2 / N
267  K = M - J: IF K < 0 THEN K = 0
268  R2 = J / M: PRINT : PRINT "NUMBER OF DATA POINTS = ";N
269  PRINT  SPC( 21)"2"
270  PRINT "DETERMINATION COEFF R = ";R2
271  PRINT "CORRELATION COEFFICIENT R = "; SQR (R2)
272  PRINT "STANDARD ESTIMATE ERROR = "; SQR (K / (N - 2))
273  RETURN
274  REM  SAVE DATA
275  PRINT : INPUT "FILENAME ? ";NM$
276  PRINT D$;"OPEN ";NM$
277  PRINT D$;"WRITE";NM$
278  PRINT N
279  FOR I = 1 TO N: PRINT X(I): PRINT Y(I): NEXT I
280  PRINT D$;"CLOSE";NM$
281  RETURN
282  REM  LOAD DATA
283  PRINT : INPUT "FILENAME ? ";NM$
284  IF NM$ = "" THEN 283
285  IF  ASC (NM$) < 65 THEN 283
286  PRINT D$;"OPEN ";NM$: PRINT D$;"READ ";NM$: INPUT N
287  FOR I = 1 TO N: INPUT X(I): INPUT Y(I): NEXT I
288  PRINT D$;"CLOSE";NM$
289  RETURN
290  HOME : INPUT "INPUT DATA POINTS FROM DISK (Y/N) ? ";ZZ$
291  IF ZZ$ <  > "Y" THEN 293
292  GOSUB 282: GOSUB 303: RETURN
293  HOME : PRINT "ENTER EACH DATA POINT (RETURN TO QUIT)": PRINT
294  PRINT  SPC( 8)"X-POINT"; SPC( 12)"Y-POINT": PRINT :N = 1
295  INPUT "X> ";XX$
296  IF XX$ = "" THEN 302
297  CALL  - 998: CALL  - 868
298  PRINT  TAB( 16 -  LEN (XX$)) VAL (XX$): CALL  - 998
299  HTAB (20): INPUT "Y> ";YY$
300  CALL  - 998: HTAB (20): CALL  - 868: HTAB (20): PRINT  SPC( 15 -  LEN
     (YY$)); VAL (YY$)
301  X(N) =  VAL (XX$):Y(N) =  VAL (YY$):N = N + 1: GOTO 295
302  N = N - 1
303  HOME : PRINT "POINT","X","Y"
304  FOR VV = 1 TO N: PRINT VV,X(VV),Y(VV): NEXT
305  PRINT : INPUT "CHANGE ANY DATA POINTS (Y/N) ";C$
306  IF  LEFT$ (C$,1) <  > "Y" THEN 310
307  GOSUB 316
308  INPUT "ANOTHER CHANGE (Y/N) ";C$
309  IF C$ = "Y" THEN 307
310  INPUT "DELETE A DATA POINT (Y/N) ";DD$
311  IF DD$ = "Y" THEN 320
312  INPUT "DO YOU WANT TO ADD DATA POINTS (Y/N) ";C$
313  IF C$ <  > "Y" THEN  RETURN
314  N = N + 1: GOTO 295
315  I = 0:J = 0:K = 0:L = 0:M = 0:Q = 0:S = 0:Z = 0:CC = 0:R2 = 0: RETURN
316  HOME : PRINT "POINT","X","Y"
317  FOR VV = 1 TO N: PRINT VV,X(VV),Y(VV): NEXT VV
318  PRINT : INPUT "CHANGE POINT # ";CC
```

Continued on next page.

Fig. 4-3 (cont). Listing for REGRESSN program.


```
319  INPUT "NEW X,Y ";X(CC),Y(CC): RETURN
320  HOME : PRINT "POINT","X","Y"
321  FOR VV = 1 TO N: PRINT VV,X(VV),Y(VV): NEXT VV
322  PRINT : INPUT "DELETE POINT # ";KK
323  FOR L = KK TO N:X(L) = X(L + 1):Y(L) = Y(L + 1): NEXT L:N = N - 1: GOTO
     303
```

Fig. 4-3 (cont). Listing for REGRESSN program.

We are then asked which point is to be changed, and then add the new coordinates for that point.

After all corrections or additions have been made, the display then informs us of the minimum and maximum limits of our data and then displays a scatter diagram (Fig. 4-5), which can then be compared against one of the generalized curves shown in Fig. 4-2. For this example, the linear, or straight line, regression appears to be the best choice. Fig. 4-6 shows the results for the linear regression line of our data, which includes the interpolation for $x = -0.171$. If no other interpolations are required, we then decide whether or not this data is to be saved on disk.

```
INPUT DATA POINTS FROM DISK (Y/N) ? N
ANTER EACH DATA POINT (RETURN TO QUIT)

          X-POINT                Y-POINT
           -2.581                   -100
           -1.101                    -20
            -.674                      0
            -.404                     13
             -.15                     25
            -.043                     30
                0                     31
            1.086                     80
            1.542                    101
            3.711                    190
X>

POINT          X                      Y
1           -2.581                 -100
2           -1.101                  -20
3           -.674                     0
4           -.404                    13
5           -.15                     25
6           -.043                    30
7           0                        31
8           1.086                    80
9           1.542                   101
10          3.711                   190

CHANGE ANY DATA POINTS (Y/N) Y
CHANGE POINT # 7
NEW X,Y 0,32
ANOTHER CHANGE (Y/N) N
DELETE A DATA POINT (Y/N) N
DO YOU WANT TO ADD DATA POINTS (Y/N) N

YOUR DATA HAS THE FOLLOWING LIMITS:

VARIABLE          MINIMUM          MAXIMUM
X                 -2.581           3.711
Y                 -100             190

HIT RETURN FOR A GRAPH OF INPUT DATA
```

Fig. 4-4. Input data for Example 4-1.

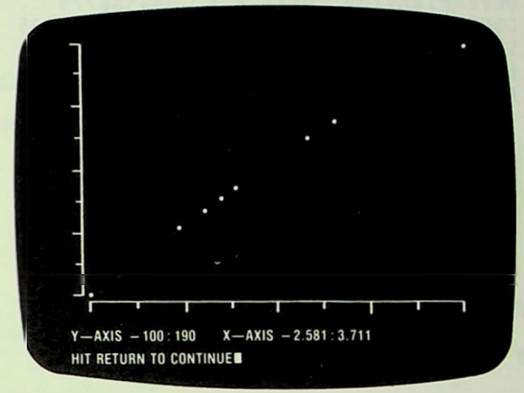


Fig. 4-5. Video scatter diagram of Example 4-1.

Because of the method in which the remaining four types of regression curves are done, the following restrictions are required:

Inverse—any x cannot equal 0.
Polynomial—order of polynomial cannot be greater than the number of data points less 2.
Geometric—all x's and y's must be greater than 0.
Exponential—any y cannot equal 0.

Table 4-1. Data for Example 4-2

Thermal Resistance (degrees/watt)	Volume (in^3)
12.0	1.0
10.0	1.3
8.0	2.0
4.0	6.0
1.3	30.0
0.8	70.0
0.4	200
0.3	300
0.13	900

```
LINEAR REGRESSION

Y = 28.7844229 + (45.5669342 * X)

NUMBER OF DATA POINTS = 10
                      2
DETERMINATION COEFF R = .995586779
CORRELATION COEFFICIENT R = .99779095
STANDARD ESTIMATE ERROR = 5.432569
INTERPOLATION (Y/N) Y
X= -.171
    THEN Y = 20.9924772
ANOTHER INTERPOLATION (Y/N) ?N
SAVE INPUT DATA TO DISK (Y/N) Y

FILENAME ? SAMPLE 1
```

Fig. 4-6. Linear regression output results.

```
POLYNOMIAL REGRESSION
                        N
Y = A  + A X +...  A X
    0      1       N

ORDER OF POLYNOMIAL 3

POLYNOMIAL COEFFICIENTS:
A(0) = 32.0932221
A(1) = 46.1803591
A(2) = -1.52832688
A(3) = .149673052

HIT ENTER TO CONTINUE
                      2
DETERMINATION COEFF R = .999966237
CORRELATION COEFFICIENT R = .999983019
STANDARD ESTIMATE ERROR = .148678266
INTERPOLATION (Y/N) Y
X = -.171
    THEN Y = 24.1509425
ANOTHER INTERPOLATION (Y/N) ?N
```

Fig. 4-7. Third-order polynomial regression output results.

Although we see that it is possible to express our data in terms of a straight line, there may be another form that is better suited. For example, it might be possible that our original data may be expressed in the form of an n-th order polynomial curve. Fig. 4-7 shows the results for the polynomial regression, using a third-order approximation. We see that the standard error is less, indicating that this might be the better form. When regression curves are compared, statistical tables should be consulted to determine whether or not the correlation coefficient is significant for the number of data points used.


```
GEOMETRIC REGRESSION (FOR X,Y>0)
      B
Y = AX

     A = 46.1540021
     B = -1.52145322

NUMBER OF DATA POINTS = 9
                      2
DETERMINATION COEFF R = .999151341
CORRELATION COEFFICIENT R = .99957558
STANDARD ESTIMATE ERROR = .0793129252
INTERPOLATION (Y/N) Y
X = 4.0
    THEN Y = 5.60031765
ANOTHER INTERPOLATION (Y/N) ?N
```

Fig. 4-8. Geometric regression results.

Example 4-2

Table 4-1 summarizes the data points from an experiment that measured the thermal resistance of a heat sink as a function of its volume. From past experience we might know that this should look like a straight line when plotted using log-log coordinates, thus suggesting a geometric regression.

Fig. 4-8 shows the output for the data of Table 4-1 using a geometric regression.

Relationship Between More Than Two Variables

The REGRESSN program dealt only with the possible relationships that may exist between a dependent variable and a single independent variable. However, there are situations where the measured quantity is dependent upon two or more quantities, which may or may not be interrelated.

The multiple linear regression program (MLREGRES) listed in Fig. 4-9 determines the coefficients for the generalized form:

$$y = a_0 + a_1x_1 + a_2x_2 \ldots + a_Nx_N$$

In addition, the program determines the coefficients of determination and correlation, as well as the standard error of estimate.

Example 4-3

For the following data, determine the multiple regression coefficients, if a solution can be found.

dependent variable y	64	71	53	67	55	58
independent variable x_1	57	59	49	62	51	50
independent variable x_2	8	10	6	11	8	7


```
100  REM  MULTIPLE LINEAR REGRESSION (MLREGRES)
101  HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "MULTIPLE LINEAR RE
     GRESSION": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR I = 1 TO 2500: NEXT I: HOME
105  DIM X(30),S(30),T(30),A(30,31)
106  PRINT : INPUT "NUMBER OF DATA POINTS? ";N
107  PRINT : INPUT "NUMBER OF INDEPENDENT VARIABLES? ";V
108  HOME
109  PRINT : PRINT "ENTER DEPENDENT & INDEPENDENT VARIABLES": PRINT "FOR
     EACH DATA POINT"
110  PRINT :X(1) = 1
111  FOR I = 1 TO N: PRINT "POINT ";I
112  INPUT "  DEPENDENT VARIABLE Y? ";X(V + 2)
113  FOR J = 1 TO V
114  PRINT "  INDEPENDENT VARIABLE A(";J;") ";
115  INPUT X(J + 1)
116  NEXT J
117  FOR K = 1 TO V + 1
118  FOR L = 1 TO V + 2
119  A(K,L) = A(K,L) + X(K) * X(L)
120  S(K) = A(K,V + 2)
121  NEXT L
122  NEXT K
123  S(V + 2) = S(V + 2) + X(V + 2) ^ 2
124  NEXT I
125  FOR I = 2 TO V + 1
126  T(I) = A(1,I)
127  NEXT I
128  FOR I = 1 TO V + 1
129  FOR J = I TO V + 1
130  IF A(J,I) <  > 0 THEN 134
131  NEXT J
132  PRINT : INVERSE : PRINT "NO UNIQUE SOLUTION": NORMAL
133  GOTO 180
134  FOR K = 1 TO V + 2
135  B = A(I,K)
136  A(I,K) = A(J,K)
137  A(J,K) = B
138  NEXT K
139  Z = 1 / A(I,I)
140  FOR K = 1 TO V + 2
141  A(I,K) = Z * A(I,K)
142  NEXT K
143  FOR J = 1 TO V + 1
144  IF J = I THEN 149
145  Z =  - A(J,I)
146  FOR K = 1 TO V + 2
147  A(J,K) = A(J,K) + Z * A(I,K)
148  NEXT K
149  NEXT J
150  NEXT I
151  HOME : PRINT : PRINT "RESULTS OF MULTIPLE LINEAR REGRESSION:": PRINT
152  PRINT "A(0) = ";A(1,V + 2)
153  FOR I = 2 TO V + 1
```

Continued on next page.

Fig. 4-9. Listing for MLREGRES program.

```
154  PRINT "A(";I - 1;") = ";A(I,V + 2)
155  NEXT I
156  P = 0
157  FOR I = 2 TO V + 1
158  P = P + A(I,V + 2) * (S(I) - T(I) * S(1) / N)
159  NEXT I
160  R = S(V + 2) - S(1) ^ 2 / N
161  Z = R - P
162  L = N - V - 1
163  I = P / V
164  I = P / R
165  PRINT : PRINT  TAB( 31)"2"
166  PRINT "COEFFICIENT OF DETERMINATION R = "; INT (I * 1E6 + .5) / 1E6
167  PRINT : PRINT "CORRELATION COEFFICIENT R = "; INT ( SQR (I) * 1E6 +
     .5) / 1E6
168  PRINT : PRINT "STANDARD ERROR OF ESTIMATE = "; INT ( SQR ( ABS (Z /
     L)) * 1E6 + .5) / 1E6
169  PRINT : INPUT "INTERPOLATION (Y/N)? ";C$
170  IF  LEFT$ (C$,1) = "Y" THEN 172
171  GOTO 180
172  P = A(1,V + 2)
173  FOR J = 1 TO V
174  PRINT : PRINT "INDEPENDENT VARIABLE A(";J;") ";
175  INPUT X
176  P = P + A(J + 1,V + 2) * X: NEXT J
177  PRINT : PRINT "  THEN Y = ";P
178  PRINT : INPUT "ANOTHER INTERPOLATION ? ";C$
179  IF C$ = "Y" THEN 172
180  PRINT : END
```

Fig. 4-9 (cont). Listing for MLREGRES program.

As shown in the output results of Fig. 4-10, the dependent variable is entered first for each point, followed by the two independent variables x_1 and x_2. The regression coefficients then yield the resulting equation:

$$y = -5.03124863 + 1.21874996x_1 - 0.0312499053x_2$$

In addition, the interpolation is shown for the values $x_1 = 52$ and $x_2 = 10$.

SAMPLE STATISTICS

The DSTATS program listed in Fig. 4-11 determines the arithmetic mean, variance, standard deviation, skewness, kurtosis, Z-scores, maximum and minimum values, and the range for a sample of ungrouped data. The variance and standard deviation is a measure of dispersion of the data about the mean value. The coefficients of skewness and kurtosis are measures of asymmetry and peakness respectively of a distribution of data. The Z-scores are measures of the deviation from the mean in terms of a standard deviation, so that other distributions of data may be compared.

When run, the DSTATS program displays the value, its difference from the mean, and Z-score in groups of 20. Hitting the RETURN key continues the rest of the list. After the entire list of values is displayed, the remaining statistics of the sample are given.

Example 4-4

Determine the sample statistics for the following 15 measured values for 100-ohm resistors: 99, 103, 99, 98, 100, 101, 105, 97, 99, 100, 102, 101, 103, 98, and 101.

As shown in Fig. 4-12, the mean value for these 15 resistors over the range of 97 to 105 is 100.4 ohms, while the variance and standard deviation are 4.828562 and 2.197399 ohms, respectively.


```
NUMBER OF DATA POINTS? 6

NUMBER OF INDEPENDENT VARIABLES? 2

ENTER DEPENDENT & INDEPENDENT VARIABLES
FOR EACH DATA POINT

POINT 1
  DEPENDENT VARIABLE Y? 64
  INDEPENDENT VARIABLE A(1) ?57
  INDEPENDENT VARIABLE A(2) ?8
POINT 2
  DEPENDENT VARIABLE Y? 71
  INDEPENDENT VARIABLE A(1) ?59
  INDEPENDENT VARIABLE A(2) ?10
POINT 3
  DEPENDENT VARIABLE Y? 53
  INDEPENDENT VARIABLE A(1) ?49
  INDEPENDENT VARIABLE A(2) ?6
POINT 4
  DEPENDENT VARIABLE Y? 67
  INDEPENDENT VARIABLE A(1) ?62
  INDEPENDENT VARIABLE A(2) 11
POINT 5
  DEPENDENT VARIABLE Y? 55
  INDEPENDENT VARIABLE A(1) ?51
  INDEPENDENT VARIABLE A(2) ?8
POINT 6
  DEPENDENT VARIABLE Y? 58
  INDEPENDENT VARIABLE A(1) ?50
  INDEPENDENT VARIABLE A(2) ?7

RESULTS OF MULTIPLE LINEAR REGRESSION:

A(0) = -5.03124863
A(1) = 1.21874996
A(2) = -.0312499053

                                  2
COEFFICIENT OF DETERMINATION R = .838158
CORRELATION COEFFICIENT R = .91551
STANDARD ERROR OF ESTIMATE = 3.696845

INTERPOLATION (Y/N)? Y

INDEPENDENT VARIABLE A(1) ?52
INDEPENDENT VARIABLE A(2) ?10
  THEN Y = 58.0312503

ANOTHER INTERPOLATION ? N
```

Fig. 4-10. Multiple linear regression results.

```
100  REM  DESCRIPTIVE STATISTICS (DSTATS)
101  CLEAR : HOME : VTAB (7): PRINT  TAB( 5): INVERSE : PRINT "DESCRIPTIV
     E STATISTICS PROGRAM": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR I = 1 TO 2500: NEXT : HOME
105  DIM N(100): FOR I = 1 TO 39:LI$ = LI$ + "=": NEXT
106  PRINT : INPUT "NUMBER OF OBSERVATIONS? ";N: PRINT
107  M = 0:B = 0:C = 0:T = 0:U = 0
108  FOR I = 1 TO N: PRINT "#";I;
109  INPUT " ";X:N(I) = X
110  C = C + X:B = B + X ^ 2: NEXT I
111  M = C / N:V = (B - N * M ^ 2) / (N - 1):T = V * (N - 1) / N:U =  SQR
     (T)
112  D =  SQR (V)
113  M =  INT (M * 1E6 + .5) / 1E6:V =  INT (V * 1E6 + .5) / 1E6:D =  INT
     (D * 1E6 + .5) / 1E6
114  PRINT : PRINT "HIT RETURN FOR LISTINGS OF": PRINT "VALUES AND Z-SCOR
     ES ";
115  CALL  - 756
116  P = 0:Q = 0:R = 0:I = 1:S = 1
117  HOME
118  HOME : PRINT : PRINT "#  VALUE    DIFFERENCE       Z-SCORE": PRINT L
     INE$
119  P = N(I) - M:Z = P / D
120  Q = Q + (P ^ 3):R = R + (P ^ 4)
121  P =  INT (P * 1E6 + .6) / 1E6:Z =  INT (Z * 1E6 + .5) / 1E6
122  PRINT I; TAB( 5)N(I); TAB( 14)P; TAB( 30)Z
123  IF I = N THEN 129
124  IF I > (20 * S) - 1 THEN 126
125  I = I + 1: GOTO 119
126  POKE  - 16368,0:S = S + 1: PRINT "  -- HIT ANY KEY TO CONTINUE -- ";
127  CALL  - 756
128  I = I + 1: GOTO 118
129  PRINT "**  END OF LIST **"
130  PRINT "HIT RETURN FOR SAMPLE STATISTICS ";
131  CALL  - 756
132  N1 = N(1):N2 = N(1)
133  FOR I = 2 TO N
134  IF (N1 - N(I)) <  = 0 THEN 136
135  N1 = N(I): GOTO 138
136  IF (N2 - N(I)) >  = 0 THEN 138
137  N2 = N(I)
138  NEXT I
139  HOME : PRINT : PRINT "MINIMUM VALUE = ";N1: PRINT : PRINT "MAXIMUM V
     ALUE = ";N2: PRINT : PRINT "RANGE = "; ABS (N2 - N1)
140  PRINT : PRINT "MEAN = ";M: PRINT : PRINT "VARIANCE = ";V: PRINT : PRINT
     "STD. DEVIATION = ";D
141  PRINT : PRINT "SKEWNESS = "; INT ((Q / (N * (U ^ 3)) * 1E6 + .5)) /
     1E6
142  PRINT : PRINT "KURTOSIS = "; INT ((R / (N * (U ^ 4)) * 1E6 + .5)) /
     1E6
143  PRINT : END
```

Fig. 4-11. Listing for DSTATS program.


```
NUMBER OF OBSERVATIONS? 15
#1 99
#2 103
#3 99
#4 98
#5 100
#6 101
#7 105
#8 97
#9 99
#10 100
#11 102
#12 101
#13 103
#14 98
#15 101

HIT RETURN FOR LISTINGS OF
VALUES AND Z SCORES


#  VALUE   DIFFERENCE       Z-SCORE
1   99      -1.4            -.637117
2   103     2.6             1.182217
3   99      -1.4            -.637117
4   98      2.4             -1.0922
5   100     -.4             -.182033
6   101     .6              .27705
7   105     4.6             2.093384
8   97      -3.4            -1.547284
9   99      -1.4            -.637117
10  400     -.4             -.182033
11  102     1.6             .728132
12  101     .6              .27705
13  103     2.6             1.182217
14  98      -2.4            -1.0922
15  101     .6              .27705

END OF LIST

MINIMUM VALUE = 97
MAXIMUM VALUE = 105
RANGE = 8
MEAN = 100.4
VARIANCE = 4.828562
STD. DEVIATION = 2.197399
SKEWNESS = .431235
KURTOSIS = 2.4851
```

Fig. 4-12. Results of sample statistics for 15 resistor values.

5

Networks and Transforms

This chapter presents programs for the design of a variety of attenuators and lossless pads, which are primarily used for impedance matching. In addition, programs are included which permit the analysis of 2-port networks, the inverse Laplace transform, real and imaginary roots of polynomials, Pi-Tee (Delta-Wye) transformations, and the solution of mesh and node equations.

ROOTS OF POLYNOMIALS

Very often the analysis of a network requires that the resultant polynomial be factored into distinct roots, which may be either real or complex. Such may be the case of solving a high order differential equation by Laplace transforms, or determining the pole locations of Butterworth and Chebyshev polynomials.

The PLYROOTS program shown in Fig. 5-1 determines the real or complex roots, if any exist, for any polynomial of the form:

$$P(x) = a_0 + a_1x + a_2x^2 \ldots + a_Nx^N \qquad \text{(Eq. 5-1)}$$

Naturally, the number of roots will be equal to the degree of the polynomial. It should be noted that third- and higher-order polynomials frequently have roots that are conjugate pairs in which the real parts are equal but the imaginary parts are of opposite sign. The coefficients of the polynomial are entered one at a time, starting with the constant term on up to the nth-order term.

Example 5-1

Determine the roots, if any exist, for the second-order polynomial:

$$P(x) = x^2 + x - 6$$

As shown in Fig. 5-2 the coefficients of the polynomial are entered one at a time, starting with the constant term on up to the second-order term. For this example, the roots were found to be equal to 2 and −3, both of which are real. This solution agrees with the solution obtained using the quadratic formula method for second-order equations.

Example 5-2

Determine the pole locations for the following fourth-order Butterworth polynomial:

$$P(s) = 1 + 2.613s + 3.414s^2 + 2.613s^3 + s^4$$

Shown in Fig. 5-3 are the output results. We see that all four roots are complex, and exist as two complex conjugate pairs. From these roots, we then know that the pole locations lie 22.5 degrees apart on the perimeter of a semicircle of unit length in the left half of the complex plane, as shown in Fig. 5-4.

THE INVERSE LAPLACE TRANSFORM

The inverse Laplace transform:

$$f(t) = \frac{1}{j2\pi}\int_{\sigma_1 - j\infty}^{\sigma_1 + j\infty} F(s)e^{st}\,ds \qquad \text{(Eq. 5-2)}$$

is frequently used for the solution of linear differential equations by transforming the problem from the frequency domain to the time varying response.

The inverse Laplace transform program (ILAPLACE) shown in Fig. 5-5 assumes that the frequency domain, or transfer function of the complex variable s ($s = j\omega$), is expressed by the following generalized form:

```
100  REM  REAL AND COMPLEX ROOTS OF A REAL POLYNOMIAL (PLYROOTS)
101  HOME : VTAB (7): PRINT  TAB( 10): INVERSE : PRINT "REAL & COMPLEX RO
     OTS": NORMAL
102  PRINT : PRINT  TAB( 10): INVERSE : PRINT "OF A REAL POLYNOMIAL": NORMAL
103  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN ,LTD."
104  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
105  FOR II = 1 TO 2500: NEXT II: HOME : DIM A(40),C(40),RR(20),RI(20)
106  PRINT : INPUT "ORDER OF POLYNOMIAL? ";O
107  PRINT : PRINT "ENTER COEFFICIENTS STARTING WITH": PRINT "THE CONSTAN
     T TERM": PRINT
108  FOR I = 1 TO O + 1
109  PRINT "COEFFICIENT A(";I - 1;") ";
110  INPUT A(I)
111  NEXT I
112  IT = O:N = O
113  IF A(N + 1) <  > O THEN 115
114  GOTO 117
115  IF N > O THEN 117
116  GOTO 206
117  NX = O
118  NY = N + 1
119  N2 = 1
120  KJ = N + 1
121  FOR L = 1 TO KJ
122  MT = KJ - L + 1
123  C(MT) = A(L)
124  NEXT L
125  REM   SET INITIAL VALUES
126  XO = .00500101:YO = .01000101
127  IN = O
128  X = XO
129  XO =  - 10 * YO:YO =  - 10 * X
130  REM   SET X AND Y TO CURRENT VALUE
131  X = XO:Y = YO
132  IN = IN + 1: GOTO 137
133  IT = 1
134  XP = X
135  YP = Y
136  REM   EVALUATE POLYNOMIAL & DERIVATIVES
137  IC = O
138  UX = O:UY = O:V = O:YT = O:XT = 1.00
139  U = C(N + 1)
140  IF U <  > O THEN 142
141  GOTO 186
142  FOR I = 1 TO N
143  L = N - I + 1
144  T = C(L)
145  X2 = X * XT - Y * YT:Y2 = X * YT + Y * XT
146  U = U + T * X2
147  V = V + T * Y2
148  F1 = I
149  UX = UX + F1 * XT * T
150  UY = UY - F1 * YT * T
151  XT = X2:YT = Y2: NEXT I
152  SQ = (UX ^ 2) + (UY ^ 2)
153  IF SQ <  > O THEN 155
```

Continued on next page.

Fig. 5-1. Listing of PLYROOTS program.


```
154  GOTO 177
155  DX = (V * UY - U * UX) / SQ
156  X = X + DX
157  DY =  - (U * UY + V * UX) / SQ
158  Y = Y + DY
159  IF  ABS (DY) +  ABS (DX) - 1E - 5 < O THEN 167
160  REM   STEP ITERATION COUNTER
161  IC = IC + 1
162  HOME : VTAB (9): PRINT  TAB( 15): FLASH : PRINT "COMPUTING": NORMAL
163  IF IC - 500 < O THEN 138
164  IF IT <  > O THEN 167
165  IF IN - 5 < O THEN 128
166  GOTO 206
167  FOR L = 1 TO NY
168  MT = KJ - L + 1
169  T = A(MT)
170  A(MT) = C(L)
171  C(L) = T: NEXT L
172  IP = N
173  N = NX
174  NX = IP
175  IF IT <  > O THEN 180
176  GOTO 133
177  IF IT <  > O THEN 179
178  GOTO 128
179  X = XP:Y = YP
180  IT = O
181  IF  ABS (Y) - (1E - 4 *  ABS (X)) < O THEN 189
182  AL = X + X
183  SQ = (X ^ 2) + Y ^ 2
184  N = N - 2
185  GOTO 193
186  X = O
187  NX = NX - 1
188  NY = NY - 1
189  Y = O
190  SQ = O
191  AL = X
192  N = N - 1
193  C(2) = C(2) + AL * C(1)
194  FOR L = 2 TO N
195  C(L + 1) = C(L + 1) + AL * C(L) - SQ * C(L - 1)
196  NEXT L
197  RI(N2) = Y:RR(N2) = X
198  N2 = N2 + 1
199  IF SQ <  > O THEN 201
200  GOTO 204
201  Y =  - Y
202  SQ = O
203  GOTO 197
204  IF N > O THEN 126
205  GOTO 209
206  INVERSE : PRINT : PRINT "UNABLE TO COMPUTE AFTER 500 ITERATIONS"
207  NORMAL
208  END
209  REM   OUTPUT RESULTS
210  HOME : PRINT : PRINT "FOR THE POLYNOMIAL EQUATION:"
211  PRINT : PRINT "O = ";A(1);" + ";A(2):"X ";
```

Continued on next page.

Fig. 5-1 (cont). Listing of PLYROOTS program.

```
212  FOR K = 3 TO 0 + 1
213  PRINT "    + ";A(K);"X^";K - 1
214  NEXT K
215  PRINT : PRINT "   THE ROOTS ARE:"
216  PRINT : PRINT "ROOT #   REAL PART      IMAGINARY PART"
217  FOR I = 1 TO 0
218  RR(I) =  INT (RR(I) * 1E6 + .5) / 1E6:RI(I) =  INT (RI(I) * 1E6 + .5)
     / 1E6
219  PRINT I; TAB( 10)RR(I); TAB( 25)RI(I): NEXT I
220  PRINT : PRINT
221  END
```

Fig. 5-1 (cont). Listing of PLYROOTS program.

```
ORDER OF POLYNOMIAL? 2

ENTER COEFFICIENTS STARTING WITH
THE CONSTANT TERM

COEFFICIENT A(0) ?-6
COEFFICIENT A(1) ?1
COEFFICIENT A(2) ?1

COMPUTING

FOR THE POLYNOMIAL EQUATION:

0 = -6 + 1X      + 1X^2

   THE ROOTS ARE:

ROOT #    REAL PART     IMAGINARY PART
1         2             0
2         -3            0
```

Fig. 5-2. Results of determining the roots for the second-order polynomial.

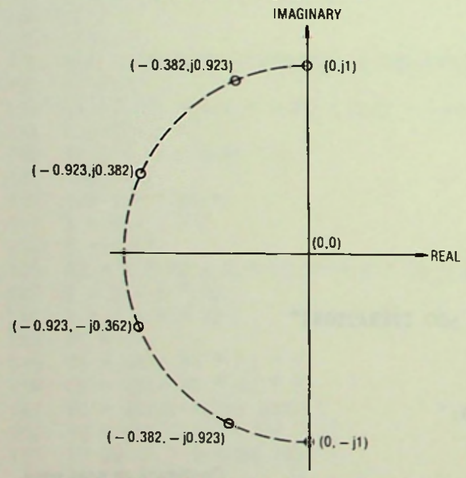


Fig. 5-4. Pole locations for fourth-order Butterworth polynomial.

$$F(s) = \frac{a_0 + a_1 s + a_2 s^2 + \ldots}{(s + b_1)(s + b_2) + \ldots} \quad \text{(Eq. 5-3)}$$

The polynomial coefficients a_i of the numerator are assumed to be real, while the denominator has been factored. Consequently the poles (i.e., the roots) of the denominator of the transfer function are known, and can be either real or complex. If this is not the case, the PLYROOT program discussed earlier in this chapter solves for both real and complex roots of a polynomial.

When executed, the program requires the number of numerator coefficients and their values, and the number of roots. Each denominator root

```
ORDER OF POLYNOMIAL? 4

ENTER COEFFICIENTS STARTING WITH
THE CONSTANT TERM

COEFFICIENT A(0) ?1
COEFFICIENT A(1) ?2.613
COEFFICIENT A(2) ?3.414
COEFFICIENT A(3) ?2.613
COEFFICIENT A(4) ?1

COMPUTING

FOR THE POLYNOMIAL EQUATION:

0 = 1 + 2.613X +   3.414X^2
    + 2.613X^3
    + 1X^4

   THE ROOTS ARE:

ROOT #    REAL PART      IMAGINARY PART
1         -.382629       -.923902
2         -.382629       .923902
3         -.923871       -.382705
4         -.923871       .382705
```

Fig. 5-3. Results of determining pole locations for the fourth-order Butterworth polynomial.


```
100  REM   INVERSE LAPLACE TRANSFORM (ILAPLACE)
101  CLEAR : HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "INVERSE LA
     PLACE TRANSFORM": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD.":
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR I = 1 TO 2500: NEXT : HOME
105  DIM A(20),CI(20),CR(20),RI(20),SR(20),SI(20),DR(20),DI(20)
106  INPUT "NUMBER OF NUMERATOR COEFFICIENTS? ";M: PRINT
107  FOR I = 1 TO M
108  PRINT "COEFFICIENT ";I;
109  INPUT "? ";A(I): NEXT I
110  PRINT : INPUT "NUMBER OF DENOMINATOR ROOTS? ";N: PRINT
111  FOR I = 1 TO N
112  PRINT "ROOT ";I;" (REAL,IMAGINARY): ";
113  INPUT "? ";RR(I),RI(I)
114  NEXT I
115  FOR I = 1 TO N
116  S1 = RR(I)
117  S2 = RI(I)
118  PR = 1
119  PI = 0
120  TR = A(1)
121  TI = 0
122  IF (M - 1) < 0 THEN 163
123  IF (M - 1) <  = 0 THEN 132
124  FOR J = 2 TO M
125  B1 = PR * S1 - PI * S2
126  C1 = PI * S1 + PR * S2
127  PR = B1
128  PI = C1
129  TR = TR + A(J) * PR
130  TI = TI + A(J) * PI
131  NEXT J
132  SR(I) = TR
133  SI(I) = TI
134  NEXT I
135  FOR J = 1 TO N
136  PR = 1
137  PI = 0
138  FOR I = 1 TO N
139  IF J - I = 0 THEN 146
140  UR = RR(J) - RR(I)
141  UI = RI(J) - RI(I)
142  B1 = PR * UR - PI * UI
143  C1 = PR * UI + PI * UR
144  PR = B1
145  PI = C1
146  NEXT I
147  DR(J) = PR
148  DI(J) = PI
149  NEXT J
150  FOR I = 1 TO N
151  CR(I) = (SR(I) * DR(I) + SI(I) * DI(I)) / ((DR(I) ^ 2) + (DI(I) ^ 2))
152  CI(I) = (SI(I) * DR(I) - SR(I) * DI(I)) / (DR(I) * DR(I) + DI(I) * DI
     (I))
153  NEXT I
```

Continued on next page.

Fig. 5-5. Listing for ILAPLACE program.

```
154  I = 1
155  PRINT : PRINT "THE INVERSE LAPLACE TRANSFORM,": PRINT "AS A FUNCTION
     OF TIME IS:": PRINT
156  PRINT "F(T) =": PRINT : PRINT "   <";CR(I);" + J(";CI(I);")> * EXP<"
     ;RR(I);
157  PRINT " +J(";RI(I);")>*T"
158  IF N <  = 1 THEN 163
159  FOR I = 2 TO N
160  PRINT : PRINT " + <";CR(I);" + J(";CI(I);")> * EXP<";RR(I);
161  PRINT " + J(";RI(I);")>*T"
162  NEXT I
163  END
```

Fig. 5-5 (cont). Listing for ILAPLACE program.

is assumed to be a complex number and is entered in the format A,B where A and B are the real and imaginary parts respectively. In addition, all roots must be distinct so that multiple roots are not allowed. The computed inverse Laplace transform time response is expressed in the form:

$$f(t) = (a_1 + jb_1)e^{(c_1 + jd_1)t} + (a_2 + jb_2)e^{(c_2 + jd_2)t} + \ldots \qquad \text{(Eq. 5-4)}$$

so that there will be as many terms as there are roots.

Example 5-3

Determine the inverse Laplace transform time response for the transfer function:

$$T(s) = \frac{s-1}{(s+2)(s+1)}$$

There are two coefficients for the numerator and two real denominator roots. Fig. 5-6 shows the

```
NUMBER OF NUMERATOR COEFFICIENTS? 2

COEFFICIENT 1? -1
COEFFICIENT 2? 1

NUMBER OF DENOMINATOR ROOTS? 2

ROOT 1 (REAL, IMAGINARY): ? -2, 0
ROOT 2 (REAL, IMAGINARY): ? -1, 0

THE INVERSE LAPLACE TRANSFORM,
AS A FUNCTION OF TIME IS:

F(T) =

   <3 + J(0)> * EXP<-2 + J(0)>*T

 + <-2 + J(0)> * EXP<-1 + J(0)>*T
```

Fig. 5-6. Results for inverse LAPLACE transfer function (Example 5-3).

```
NUMBER OF NUMERATOR COEFFICIENTS? 1

COEFFICIENT 1? 1

NUMBER OF DENOMINATOR ROOTS? 2

ROOT 1 (REAL, IMAGINARY): ? -2, -1
ROOT 2 (REAL, IMAGINARY): ? -2, 1

THE INVERSE LAPLACE TRANSFORM,
AS A FUNCTION OF TIME IS:

F(T) =

   <0 + J(.5)> * EXP<-2 + J(-1)>*T

 + <0 + J(-.5)> * EXP<-2 + J(1)>*T
```

Fig. 5-7. Results for inverse LAPLACE transfer function (Example 5-4).

output results for this transfer function which has the time variant solution:

$$f(t) = 3e^{-2t} - 2e^{-t}$$

which agrees with the result obtained by the use of inverse Laplace transform tables.

Example 5-4

Determine the inverse Laplace transform time response for the transfer function:

$$T(s) = \frac{1}{s^2 + 4s + 5} = \frac{1}{(s+2+j)(s+2-j)}$$

For this transfer function, there are 2 complex roots which are complex conjugates. The output results are shown in Fig. 5-7, but can be simplified further.

The computed solution can then be rewritten as:

$$j0.5e^{-5t}e^{-(2+j)t} - j0.5e^{-(2-j)t} = e^{-2t}(0.5je^{-t} - 0.5je^{t}) = e^{-2t}\sin(t) \qquad \text{(Eq. 5-5)}$$


```
NUMBER OF NUMERATOR COEFFICIENTS? 1

COEFFICIENT 1? 1

NUMBER OF DENOMINATOR ROOTS? 3

ROOT 1 (REAL, IMAGINARY): ? 0, 0
ROOT 2 (REAL, IMAGINARY): ? -1, 0
ROOT 3 (REAL, IMAGINARY): ? 1, 0

THE INVERSE LAPLACE TRANSFORM,
AS A FUNCTION OF TIME IS:

F(T) =

   <-1 + J(0)> * EXP<0 + J(0)>*T

 + <.5 + J(0)> * EXP<-1 + J(0)>*T

 + <.5 + J(0)> * EXP<1 + J(0)>*T
```

Fig. 5-8. Results for inverse LAPLACE transfer function (Example 5-5).

using the exponential identities of trigonometric functions.

Example 5-5

Determine the inverse Laplace transform time response for the transfer function:

$$T(s) = \frac{1}{s(s^2-1)} = \frac{1}{s(s+1)(s-1)}$$

Fig. 5-8 shows the output results, which can be simplified further. Using the identity:

$$\frac{e^t + e^{-t}}{2} = \cosh(t) \qquad \text{(Eq. 5-6)}$$

so that the computed answer can now be more formally stated as:

$$f(t) = \cosh(t) - 1$$

which can be verified using tables of inverse Laplace transforms.

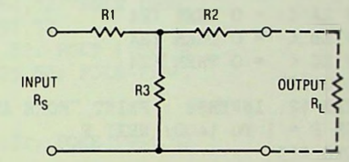
IMPEDANCE MATCHING PADS

Impedance matching devices are frequently used to "match" the signal source to a given load impedance. These devices may be either transformers, transistor circuits, or a class of networks called pads. Pads may be composed solely of resistors, or they may be a combination of inductors and capacitors. In this section, programs for the design of several common types of resistive and lossless matching pads are discussed.

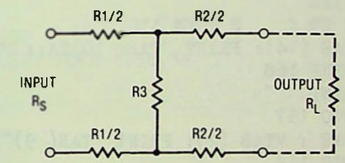
Resistance Pads

A resistance pad is a 4-terminal resistor network that permits the matching of a signal source to a load, but with a finite amount of loss, or attenuation. This is because resistors are passive devices and, unlike a transistor circuit or amplifier, cannot produce gain. The four main types of resistance pad networks are shown in Fig. 5-9. The T- and Pi-networks are unbalanced pads while the H- and O-pads are their respective balanced counterparts. When the source and load impedances are equal, these networks are referred to as attenuators, otherwise they are called pads when the source and load impedances are not equal.

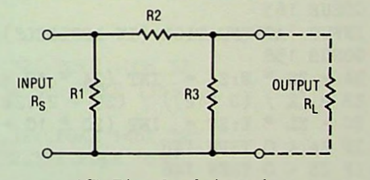
The ATTNPAD program shown in Fig. 5-10 is a graphics program to determine the required pad resistances when the source and load impedances, as well as the required amount of attenuation (in decibels), are known. The final schematic is included as part of the video display.



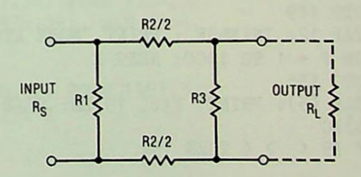
(A) T-pad unbalanced.



(B) H-pad balanced.



(C) Pi-pad unbalanced.



(D) O-pad balanced.

Fig. 5-9. Resistive matching pads.

```
100  REM   ATTENUATOR PAD DESIGN WITH GRAPHICS (ATTNPAD)
101  HOME : VTAB (7): PRINT  TAB( 9): INVERSE : PRINT "ATTENUATOR PAD DES
     IGN": NORMAL : PRINT : PRINT  TAB( 13): INVERSE : PRINT "WITH GRAPHI
     CS": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II
105  HOME : PRINT "ATTENUATOR DESIGN:"
106  CLEAR
107  PRINT : PRINT "   1. T-PAD (UNBALANCED)"
108  PRINT : PRINT "   2. H-PAD (BALANCED)"
109  PRINT : PRINT "   3. PI-PAD (UNBALANCED)"
110  PRINT : PRINT "   4. O-PAD (BALANCED)"
111  POKE  - 16368,0: PRINT : PRINT "   5. QUIT": PRINT : INPUT "SELECT #
     ? ";Z9: IF Z9 < 1 OR Z9 > 5 THEN 105
112  ON Z9 GOTO 113,133,135,155,999
113  HOME : VTAB (2): PRINT  TAB( 9)"UNBALANCED T-PAD DESIGN"
114  GOSUB 163
115  INPUT "ATTENUATION (IN DECIBELS): ?";N
116  GOSUB 158
117  ZA = (ZS * C - 2 * X * D) / Z9:ZA =  INT (ZA * 10 + .5) / 10
118  ZB = (ZL * C - 2 * X * D) / Z9:ZB =  INT (ZB * 10 + .5) / 10
119  ZC = 2 * X * D:ZC =  INT (ZC * 10 + .5) / 10
120  IF ZA <  = 0 THEN 124
121  IF ZB <  = 0 THEN 124
122  IF ZC <  = 0 THEN 124
123  GOTO 127
124  VTAB 12: INVERSE : PRINT "MORE ATTENUATION IS REQUIRED, TRY AGAIN": NORMAL
125  FOR F = 1 TO 1400: NEXT F
126  GOTO 113
127  VTAB (5): PRINT  TAB( 10)ZA;: PRINT  TAB( 28)ZB: VTAB (12): PRINT  TAB(
     25)ZC
128  IF Z9 <  > 2 THEN 130
129  VTAB (14): PRINT  TAB( 10)ZA;: PRINT  TAB( 28)ZB
130  GOSUB 168
131  PRINT " "
132  GOTO 157
133  HOME : VTAB (2): PRINT  TAB( 9)"BALANCED H-PAD DESIGN"
134  GOTO 114
135  HOME : VTAB (2): PRINT  TAB( 8)"UNBALANCED PI-PAD DESIGN"
136  GOSUB 163
137  INPUT "ATTENUATION (IN DECIBELS): ?";N
138  GOSUB 158
139  ZA = ZS * E:ZA =  INT (ZA * 10 + .5) / 10
140  ZB = (X / (D * 2)) / (Z9 - 2):ZB =  INT (ZB * 10 + .5) / 10
141  ZC = ZL * E:ZC =  INT (ZC * 10 + .5) / 10
142  IF ZA < 0 THEN 146
143  IF ZB < 0 THEN 146
144  IF ZC < 0 THEN 146
145  GOTO 149
146  VTAB 12: INVERSE : PRINT "MORE ATTENUATION REQUIRED, TRY AGAIN": NORMAL
147  FOR F = 1 TO 1400: NEXT F
148  GOTO 135
149  VTAB (5): PRINT  TAB( 19)ZB: VTAB (15): PRINT  TAB( 7)ZA;: PRINT  TAB(
     28)ZC
150  IF Z9 <  > 4 THEN 152
```

Continued on next page.

Fig. 5-10. Listing for ATTNPAD program.


```
151  VTAB (19): PRINT  TAB( 19)ZB
152  GOSUB 181
153  PRINT " "
154  GOTO 157
155  HOME : VTAB (2): PRINT  TAB( 9)"BALANCED O-PAD DESIGN"
156  GOTO 136
157  VTAB 24: POKE  - 16368,0: CALL  - 756: GOTO 105: REM           BACK
     TO BEGINNING
158  K2 = 10 ^ (N / 10):K =  SQR (K2)
159  C = (K2 + 1) / (K2 - 1):D = K / (K2 - 1)
160  Z =  SQR (ZS / ZL):X =  SQR (ZL * ZS)
161  E = (K2 - 1) / (K2 - (2 * K / Z) + 1)
162  RETURN
163  VTAB (20): PRINT "INPUT IMPEDANCE: ";
164  INPUT ZS
165  PRINT "LOAD IMPEDANCE: ";
166  INPUT ZL: RETURN
167  VTAB (23): END
168  VTAB (12): PRINT "INPUT": VTAB (13): PRINT  TAB( 35)"LOAD"
169  POKE 1805,32: POKE 1806,32: POKE 1807,32: POKE 1808,32: POKE 1816,32
     : POKE 1817,32: POKE 1818,32: POKE 1819,32: POKE 1930,32: POKE 1931,
     32: POKE 1932,32: POKE 1933,32
170  FOR I = 1936 TO 1944: POKE I,32: NEXT I
171  POKE 1947,32: POKE 1948,32: POKE 1949,32: POKE 1950,32: POKE 1077,32
     : POKE 1078,32: POKE 1079,32: POKE 1080,32: POKE 1084,32: POKE 1088,
     32: POKE 1089,32: POKE 90,32: POKE 1090,32: POKE 1091,32
172  POKE 1212,32: POKE 1339,32: POKE 1340,32: POKE 1341,32: POKE 1467,32
     : POKE 1469,32: POKE 1595,32: POKE 1597,32: POKE 1723,32: POKE 1724,
     32: POKE 1725,32: POKE 1852,32: POKE 1980,32
173  IF Z9 <  > 2 THEN 179
174  POKE 1973,32: POKE 1974,32: POKE 1975,32: POKE 1976,32: POKE 1984,32
     : POKE 1985,32: POKE 1986,32: POKE 1987,32
175  POKE 1114,32: POKE 1115,32: POKE 1116,32: POKE 1117,32: FOR I = 1120
      TO 1128: POKE I,32: NEXT I
176  POKE 1131,32: POKE 1132,32: POKE 1133,32: POKE 1134,32
177  POKE 1245,32: POKE 1246,32: POKE 1247,32: POKE 1248,32: POKE 1256,32
     : POKE 1257,32: POKE 1258,32: POKE 1259,32
178  RETURN
179  FOR I = 1114 TO 1134: POKE I,32: NEXT I
180  RETURN
181  VTAB (12): PRINT "INPUT": VTAB (13): PRINT  TAB( 35)"LOAD"
182  POKE 1810,32: POKE 1811,32: POKE 1812,32: POKE 1813,32
183  FOR I = 1931 TO 1938: POKE I,32: NEXT I
184  FOR I = 1941 TO 1949: POKE I,32: NEXT I
185  POKE 1079,32: POKE 1082,32: POKE 1083,32: POKE 1084,32: POKE 1085,32
     : POKE 1088,32: POKE 1207,32: POKE 1216,32: POKE 1334,32: POKE 1335,
     32: POKE 1336,32
186  POKE 1343,32: POKE 1344,32: POKE 1345,32: POKE 1462,32: POKE 1464,32
     : POKE 1471,32: POKE 1473,32: POKE 1590,32: POKE 1592,32: POKE 1599,
     32: POKE 1601,32
187  POKE 1718,32: POKE 1719,32: POKE 1720,32: POKE 1727,32: POKE 1728,32
     : POKE 1729,32: POKE 1847,32: POKE 1856,32: POKE 1975,32: POKE 1984,
     32
188  IF Z9 <  > 4 THEN 193
189  POKE 1975,32: POKE 1978,32: POKE 1979,32: POKE 1980,32: POKE 1981,32
     : POKE 1984,32: FOR I = 1114 TO 1122: POKE I,32: NEXT I
190  FOR I = 1125 TO 1132: POKE I,32: NEXT I
191  FOR I = 1250 TO 1253: POKE I,32: NEXT I
```

Continued on next page.

Fig. 5-10 (cont). Listing for ATTNPAD program.

```
192  RETURN
193  FOR I = 1114 TO 1134: POKE I,32: NEXT I
194  RETURN
999  HOME : VTAB 12: PRINT "GOODBYE....."
```

Fig. 5-10 (cont). Listing for ATTNPAD program.

Depending on the relationship between the source and load impedances, there is usually a minimum value of attenuation required in order to satisfy the design. If more attenuation is required, the program informs us of that fact.

Example 5-6

Design a T-pad having an attenuation of 10 dB while matching a 300-ohm source to a 75-ohm load. When run, we find that it is not possible to have 10-dB attenuation. Consequently, we must choose a higher value, such as 15 dB. Fig. 5-11 shows the final circuit required.

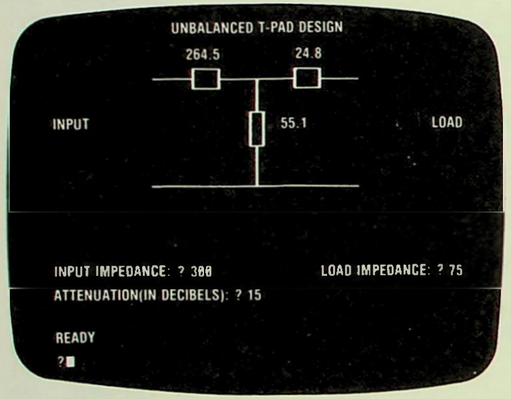


Fig. 5-11. Video results of T-pad having an attenuation of 15 dB.

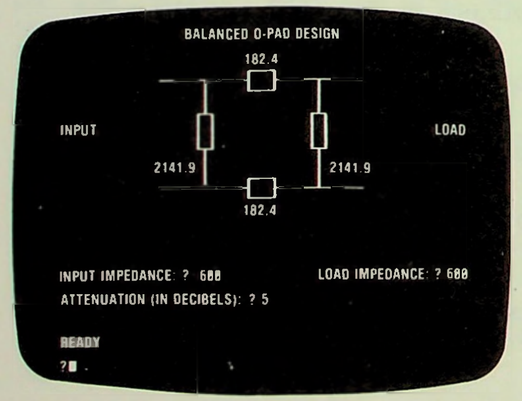


Fig. 5-12. Video results of O-pad having an attenuation of 5 dB.

Example 5-7

Design an O-pad having an attenuation of 5 dB between a 600-ohm source and load. Fig. 5-12 shows the final circuit required.

Lossless Pads

Fig. 5-13 shows two types of lossless pads that are used to match impedances at a single frequency. Both circuits are equivalent, and require that the source impedance be less than the load impedance. If the source impedance is greater than the load, then either circuit is turned around so that the shunt element is now in parallel with the source instead of with the load.

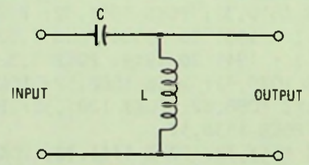
Fig. 5-14 lists the LCPAD program which determines the required series capacitor/shunt inductor circuit (Fig. 5-13A), or the equivalent series inductor/shunt capacitor circuit (Fig. 5-13B) when the source and load impedances, as well as the input frequency, are known. The LC-PAD program includes a schematic which is included as part of the video display.

Example 5-8

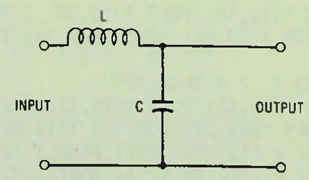
Design a lossless pad to match a 50-ohm source to a 75-ohm load at 3.5 MHz.

The two possible circuits are shown in Fig. 5-15, along with the required values.

Another version of the lossless pad is to be able to specify the phase shift in electrical degrees.



(A) Series capacitor/shunt inductor.



(B) Series inductor/shunt capacitor.

Fig. 5-13. LC lossless pads for a specific frequency.


```
100  REM   LOSSLESS L-C PAD DESIGN - GRAPHICS (LCPAD)
101  HOME : VTAB (7): PRINT  TAB( 9): INVERSE : PRINT "LOSSLESS L-C PAD D
     ESIGN": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II
105  HOME : PRINT : PRINT  TAB( 10)"LOSSLESS L-C PAD DESIGN"
106  VTAB (21): PRINT "INPUT IMPEDANCE ";
107  INPUT Z1
108  PRINT "LOAD IMPEDANCE ";
109  INPUT Z2: IF Z2 < Z1 THEN 105
110  INPUT "INPUT FREQUENCY (IN HZ)? ";F
111  W = F / 6.28318
112  X1 = Z2 *  SQR (Z1 / (Z2 - Z1)):X2 =  SQR (Z1 * (Z2 - Z1))
113  S1 = 1 * 1E9 / (W * X2):H1 = X1 * 1E6 / W
114  S1 =  INT (S1 * 100 + .5) / 100:H1 =  INT (H1 * 100 + .5) / 100
115  S2 = X2 * 1E6 / W:H2 = 1 * 1E9 / (W * X1)
116  S2 =  INT (S2 * 100 + .5) / 100:H2 =  INT (H2 * 100 + .5) / 100
117  VTAB (4): PRINT  TAB( 11)S1;" NF"; TAB( 32)S2;" UH"
118  VTAB (15): PRINT  TAB( 13)H1;" UH": VTAB (13): PRINT  TAB( 25)H2;" N
     F"
119  VTAB (10): PRINT  TAB( 20)"OR";
120  FOR I = 1 TO 1000: READ X: IF X = 999 GOTO 130
121  POKE X,32: NEXT I
122  DATA  1540,1542,1562,1563,1564,1565,1566,1567,1568
123  DATA  1665,1666,1667,1668,1670,1671,1672,1673,1674,1675,1676,1677,16
     87,1688,1689,1690,1692,1694,1696,1697,1698,1699,1700,1701,1702
124  DATA  1796,1798,1801,1820,1822,1826,1929,1930,1954
125  DATA  1074,1098,1200,1201,1202,1225,1226,1227,1330
126  DATA  1456,1457,1458,1481,1482,1483,1586,1610
127  DATA  1713,1714,1738,1841,1866,1969,1994
128  DATA  1104,1105,1106,1107,1108,1109,1110,1111,1112,1113,1114,1115,11
     16,1117,1127,1128,1129,1130,1131,1132,1133,1134,1135,1136,1137,1138,
     1139,1140,1141,1142
129  DATA  999
130  VTAB (23): END
```

Fig. 5-14. Listing for LCPAD program.

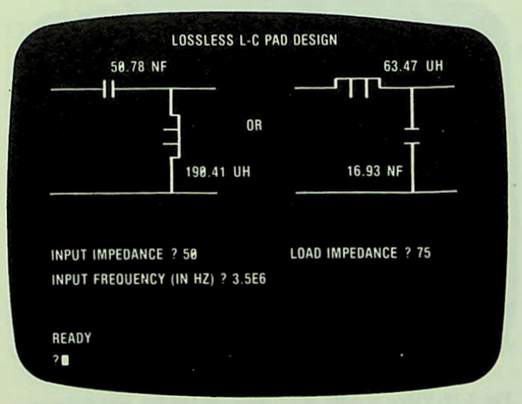


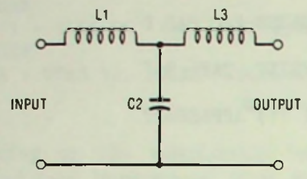
Fig. 5-15. Video results of lossless LC pad design.

Depending on the source and load impedances, and the amount of phase shift required, there are four circuits possible, as shown in Fig. 5-16. The convention is taken that if the input signal is to lead the output, the phase angle is positive; otherwise, it is negative if the input lags the output.

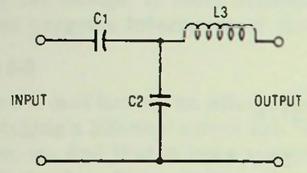
The LCPSPAD program of Fig. 5-17 is used for designing these types of lossless pads. Like ATTNPAD and LCPAD, this program includes the final schematic with component values as part of the video display.

Example 5-9

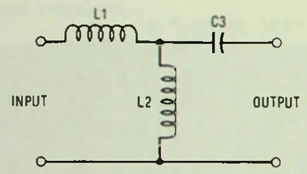
Design a lossless pad to match a 50-ohm source with a 75-ohm load with a phase shift of +45 degrees at 3.5 MHz. Repeat this example, but for a phase shift of −45 degrees.



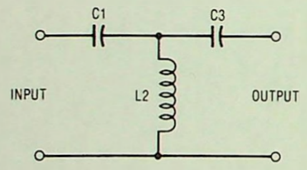
(A) Large lag section.



(B) Small lag section.



(C) Large lead section.



(D) Small lead section.

Fig. 5-16. LC lossless pads for a specific phase shift.

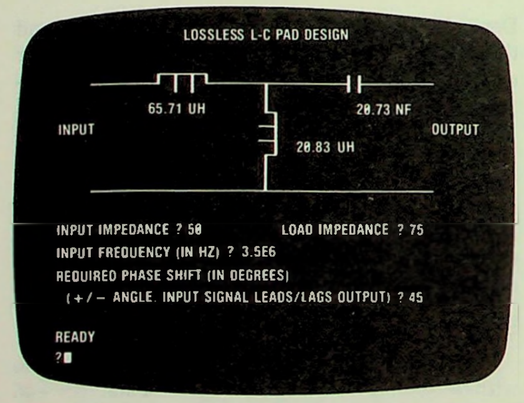


Fig. 5-18. Video results of LC lossless pad for +45 degrees phase shift.

The final circuit for +45 degrees phase shift is shown in Fig. 5-18, while the display of Fig. 5-19 shows the required circuit having a phase shift of −45 degrees.

PI-TEE (DELTA-WYE) TRANSFORMS

Networks often topologically appear in forms resembling either "Tee" or "Pi" shaped networks (sometimes called Wye or Delta networks), as shown in Fig. 5-20. Examples are the resistive attenuators discussed previously, Wheatstone bridge, and lattice circuits. The ability to transform from Tee to Pi, or vice versa, simplifies the analysis of a number of circuits.

The TEEPI program listed in Fig. 5-21 allows the user to determine the corresponding equivalent network. The starting and equivalent schematics are displayed in the screen along with their values.

Example 5-10

For the unbalanced T-pad design of Example 5-6, determine the equivalent Pi (or Delta) circuit.

From the results of Example 5-6, Z1, Z2, and Z3 are 264.5, 55.1, and 24.8 ohms respectively. The equivalent Pi circuit is shown in Fig. 5-22. In fact, the resulting Pi circuit should be the same as if we had chosen to design an unbalanced Pi-pad in Example 5-6 instead of a T-pad.

MESH CURRENT NETWORK ANALYSIS

In an electric network, mesh currents (or loop currents) are produced in each circuit branch as a result of one or more applied voltage sources. From these mesh currents, one can then determine

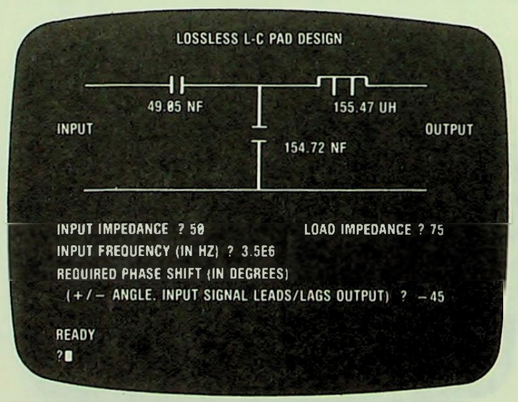


Fig. 5-19. Video results of LC lossless pad for −45 degrees phase shift.


```
100  REM  LOSSLESS L-C PAD WITH PHASE SHIFT (WITH GRAPHICS)  LCPSPAD
101  HOME : VTAB (7): PRINT  TAB( 8): INVERSE : PRINT "LOSSLESS L-C PAD D
     ESIGN": NORMAL : PRINT : PRINT  TAB( 12): INVERSE : PRINT "WITH PHAS
     E SHIFT": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II
105  HOME : VTAB (2): PRINT  TAB( 7)"LOSSLESS L-C PAD DESIGN"
106  VTAB (17): PRINT "INPUT IMPEDANCE?";
107  INPUT " ";Z1
108  PRINT "LOAD IMPEDANCE?";
109  INPUT " ";Z2
110  INPUT "INPUT FREQUENCY (IN HZ)? ";F
111  PRINT "REQUIRED PHASE SHIFT IN DEGREES "
112  PRINT " (+/- ANGLE": INPUT "SIGNAL LEADS/LAGS OUTPUT)?";D
113  W = F / 6.28319:R = D / 57.2957
114  Z = Z1 / Z2:Y =  SQR (Z)
115  A = Z *  SIN (R) / (Y -  COS (R))
116  B = Y *  SIN (R)
117  C = Y *  SIN (R) / (1 - Y *  COS (R))
118  IF R < 0 THEN 120
119  IF R > 0 THEN 137: GOTO 118
120  X3 = Z1 /  ABS (B):L3 = X3 * 1E6 / W
121  X2 = Z1 /  ABS (A):C2 = 1E9 / (W * X2)
122  L3 =  INT (L3 * 100 + .5) / 100
123  C2 =  INT (C2 * 100 + .5) / 100
124  IF C < 0 THEN 126
125  IF C > 0 THEN 132: GOTO 124
126  X1 = Z1 /  ABS (C):C1 = 1E9 / (W * X1)
127  C1 =  INT (C1 * 100 + .5) / 100
128  L3 =  INT (L3 * 100 + .5) / 100
129  VTAB (8): PRINT  TAB( 6)C1;" NF";: PRINT  TAB( 25)L3;" UH"
130  VTAB (10): PRINT " INPUT";: PRINT  TAB( 33)"OUTPUT": VTAB (12): PRINT
     TAB( 23)C2;" NF"
131  GOSUB 153: GOTO 152
132  X1 = Z1 / C:L1 = X1 * 1E6 / W
133  L1 =  INT (L1 * 100 + .5) / 100
134  VTAB (8): PRINT  TAB( 6)L1;" UH";: PRINT  TAB( 25)L3;" UH"
135  VTAB (10): PRINT " INPUT";: PRINT  TAB( 33)"OUTPUT": VTAB (12): PRINT
     TAB( 23)C2;" NF"
136  GOSUB 161: GOTO 152
137  X2 = Z1 / A:L2 = X2 * 1E6 / W:X3 = Z1 / B:C3 = 1E9 / (W * X3)
138  L2 =  INT (L2 * 100 + .5) / 100
139  C3 =  INT (C3 * 100 + .5) / 100
140  IF C < 0 THEN 142
141  IF C > 0 THEN 147: GOTO 140
142  X1 = Z1 /  ABS (C):C1 = 1E9 / (W * X1)
143  C1 =  INT (C1 * 100 + .5) / 100
144  VTAB (8): PRINT  TAB( 6)C1;" NF";: PRINT  TAB( 25)C3;" NF"
145  VTAB (10): PRINT "INPUT";: PRINT  TAB( 33)"OUTPUT": VTAB (12): PRINT
     TAB( 23)L2;" UH"
146  GOSUB 169: GOTO 152
147  X1 = Z1 / C:L1 = X1 * 1E6 / W
148  L1 =  INT (L1 * 100 + .5) / 100
149  VTAB (8): PRINT  TAB( 6)L1;" UH";: PRINT  TAB( 25)C3;" NF"
```

Continued on next page.

Fig. 5-17. Listing for LCPSPAD program.

```
150  VTAB (10): PRINT "INPUT";: PRINT  TAB( 33)"OUTPUT": VTAB (12): PRINT
     TAB( 23)L2;" UH"
151  GOSUB 177
152  VTAB (23): END
153  POKE 1417,32: POKE 1419,32: FOR I = 1432 TO 1438: POKE I,32: NEXT I
154  FOR I = 1540 TO 1545: POKE I,32: NEXT I
155  FOR I = 1547 TO 1560: POKE I,32: NEXT I
156  POKE 1562,32: POKE 1564,32: FOR I = 1566 TO 1570: POKE I,32: NEXT I
157  POKE 1673,32: POKE 1675,32: POKE 1682,32: POKE 1690,32: POKE 1692,32
     : POKE 1810,32: POKE 1818,32: POKE 1820,32: POKE 1938,32: POKE 1081,
     32: POKE 1082,32: POKE 1083,32
158  POKE 1337,32: POKE 1338,32: POKE 1339,32: POKE 1466,32: POKE 1594,32
     : POKE 1722,32
159  FOR I = 1836 TO 1866: POKE I,32: NEXT I
160  RETURN
161  FOR I = 1415 TO 1421: POKE I,32: NEXT I: FOR I = 1432 TO 1438: POKE
     I,32: NEXT I
162  FOR I = 1540 TO 1543: POKE I,32: NEXT I: POKE 1545,32: POKE 1547,32
163  FOR I = 1549 TO 1560: POKE I,32: NEXT I: POKE 1562,32: POKE 1564,32:
      FOR I = 1566 TO 1569: POKE I,32: NEXT I
164  POKE 1673,32: POKE 1675,32: POKE 1683,32: POKE 1690,32: POKE 1692,32
165  POKE 1811,32: POKE 1939,32: POKE 1082,32: POKE 1083,32: POKE 1084,32
     : POKE 1338,32: POKE 1339,32: POKE 1340,32
166  POKE 1467,32: POKE 1595,32: POKE 1723,32
167  FOR I = 1835 TO 1865: POKE I,32: NEXT I
168  RETURN
169  POKE 1417,32: POKE 1419,32: POKE 1434,32: POKE 1436,32: FOR I = 1539
      TO 1545: POKE I,32: NEXT I
170  FOR I = 1547 TO 1562: POKE I,32: NEXT I
171  FOR I = 1564 TO 1568: POKE I,32: NEXT I
172  POKE 1673,32: POKE 1675,32: POKE 1683,32: POKE 1690,32: POKE 1692,32
     : POKE 1811,32: POKE 1812,32: POKE 1940,32: POKE 1081,32: POKE 1082,
     32: POKE 1083,32: POKE 1084,32
173  POKE 1212,32: FOR I = 1337 TO 1340: POKE I,32: NEXT I
174  POKE 1468,32: POKE 1595,32: POKE 1596,32: POKE 1723,32
175  FOR I = 1835 TO 1865: POKE I,32: NEXT I
176  RETURN
177  FOR I = 1415 TO 1421: POKE I,32: NEXT I
178  POKE 1434,32: POKE 1436,32: POKE 1540,32: POKE 1541,32: POKE 1542,32
     : POKE 1543,32: POKE 1545,32: POKE 1547,32
179  FOR I = 1549 TO 1562: POKE I,32: NEXT I: FOR I = 1564 TO 1569: POKE
     I,32: NEXT I
180  POKE 1673,32: POKE 1675,32: POKE 1683,32: POKE 1690,32: POKE 1692,32
     : POKE 1811,32: POKE 1812,32
181  POKE 1940,32: POKE 1081,32: POKE 1082,32: POKE 1083,32: POKE 1084,32
     : POKE 1212,32: FOR I = 1337 TO 1340: POKE I,32: NEXT I
182  POKE 1468,32: POKE 1595,32: POKE 1596,32: POKE 1723,32
183  FOR I = 1835 TO 1865: POKE I,32: NEXT I
184  RETURN
```

Fig. 5-17 (cont). Listing for LCPSPAD program.

the voltage across or the current through any network element.

Once the mesh equations are established, the individual mesh currents are found by solving a set of simultaneous equations. Fig. 5-23 shows the SIMULEQU program listing to solve a set of simultaneous equations having real coefficients if a solution is possible. When run, the program requires the number of equations, which is the same number of unknown variables and the minimum allowable value for any solution, which usually is set at 1E-6 (10^{-6}). The coefficients are then entered, one at a time for each equation, with the constant term entered last. For the equation: $9X - 8Y + 3Z = 5$, the "9" is the first coefficient, while the "5" is the fourth coefficient.

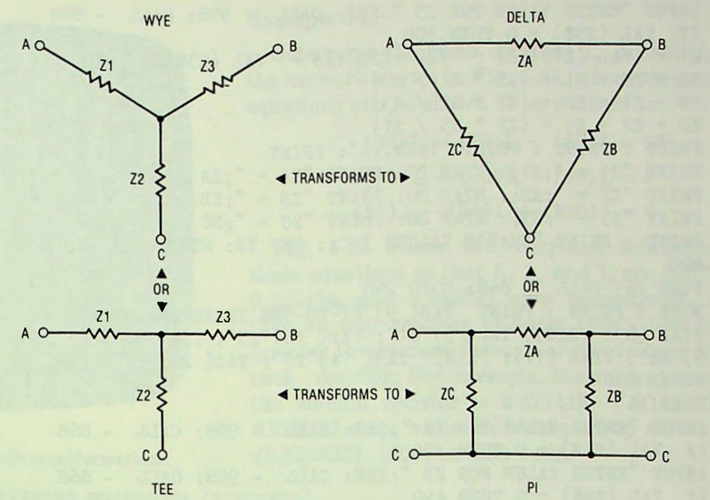


Fig. 5-20. Pi-Tee and Delta-Wye equivalent networks.

```
100  REM  PI-TEE (DELTA-WYE) TRANSFORM -(TEEPI)
110  HOME
120  VTAB (7): PRINT  TAB( 11): INVERSE : PRINT "PI-TEE (DELTA-WYE)": NORMAL
130  PRINT : PRINT  TAB( 13): INVERSE : PRINT "TRANSFORMATION": NORMAL : PRINT
     : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"HOWARD
     M. BERLIN, LTD."
140  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
150  FOR II = 1 TO 2500: NEXT II: HOME
160  PRINT : PRINT "SELECT:"
170  PRINT "      (T)EE-PI"
180  PRINT : PRINT "      (P)I-TEE"
190  PRINT : INPUT "INPUT CHOICE (T OR P)? ";C$
200  IF C$ < > "T" AND C$ < > "P" THEN 190
210  IF C$ = "P" THEN 430
220  HOME : PRINT : PRINT  TAB( 8)"TEE-TO-PI TRANSFORMATION"
230  VTAB (8): PRINT  TAB( 5)"Z1"; TAB( 16)"Z3"; TAB( 19)"TO-->"; TAB( 29
     )"ZA": VTAB (11): PRINT  TAB( 14)"Z2"; TAB( 22)"ZC"; TAB( 38)"ZB"
240  GOSUB 650
250  VTAB (16)
260  INPUT "ENTER VALUE FOR Z1 ";Z1$: CALL  - 998: CALL  - 868
270  IF  VAL (Z1$) = 0 THEN 260
280  INPUT "ENTER VALUE FOR Z2 ";Z2$: CALL  - 998: CALL  - 868
290  IF  VAL (Z2$) = 0 THEN 280
```

Continued on next page.

Fig. 5-21. Listing for TEEPI program.

```
300  INPUT "ENTER VALUE FOR Z3 ";Z3$: CALL  - 998: CALL  - 868
310  IF  VAL (Z3$) = 0 THEN 300
320  Z1 =  VAL (Z1$):Z2 =  VAL (Z2$):Z3 =  VAL (Z3$)
340  ZA = Z3 + Z1 + (Z3 * Z1 / Z2)
350  ZB = Z1 + Z2 + (Z1 * Z2 / Z3)
360  ZC = Z2 + Z3 + (Z2 * Z3 / Z1)
370  PRINT : PRINT : PRINT "THEN...": PRINT
380  PRINT "Z1 = ";Z1;: HTAB 20: PRINT "ZA = ";ZA
390  PRINT "Z2 = ";Z2;: HTAB 20: PRINT "ZB = ";ZB
400  PRINT "Z3 = ";Z3;: HTAB 20: PRINT "ZC = ";ZC
410  PRINT : PRINT "CHANGE VALUES ? ";: GET Y$: HTAB 1: IF Y$ <  > "Y" THEN
     800
420  VTAB 16: CALL  - 868: GOTO 250
430  HOME : PRINT : PRINT  TAB( 9)"PI-TO-TEE TRANSFORMATION"
440  VTAB (8): PRINT  TAB( 5)"Z1"; TAB( 16)"Z3"; TAB( 18)" <--TO"; TAB( 2
     9)"ZA": VTAB (11): PRINT  TAB( 14)"Z2"; TAB( 22)"ZC"; TAB( 38)"ZB"
450  GOSUB 650
460  VTAB 16
470  INPUT "ENTER VALUE FOR ZA ";ZA$: CALL  - 998: CALL  - 868
480  IF  VAL (ZA$) = 0 THEN 470
490  INPUT "ENTER VALUE FOR ZB ";ZB$: CALL  - 998: CALL  - 868
500  IF  VAL (ZB$) = 0 THEN 490
510  INPUT "ENTER VALUE FOR ZC ";ZC$: CALL  - 998: CALL  - 868
520  IF  VAL (ZC$) = 0 THEN 510
530  ZA =  VAL (ZA$):ZB =  VAL (ZB$):ZC =  VAL (ZC$)
540  Z = ZA + ZB + ZC
550  Z1 = ZB * ZA / Z
560  Z2 = ZB * ZC / Z
570  Z3 = ZA * ZC / Z
580  PRINT : PRINT : PRINT "THEN...": PRINT
590  PRINT "ZA = ";ZA;: HTAB 20: PRINT "Z1 = ";Z1
600  PRINT "ZB = ";ZB;: HTAB 20: PRINT "Z2 = ";Z2
610  PRINT "ZC = ";ZC;: HTAB 20: PRINT "Z3 = ";Z3
620  PRINT : PRINT "CHANGE VALUES ? ";: GET Y$: HTAB 1: IF Y$ <  > "Y" THEN
     800
630  VTAB 16: CALL  - 868: GOTO 460
640  GOTO 460
650  FOR I = 1 TO 150
660  READ X
670  IF X = 99 THEN 700
680  POKE X,32
690  NEXT I
700  RETURN
710  DATA  1412,1413,1414,1415,1421,1422,1423,1424,1436,1437,1438,1439
720  DATA  1538,1539,1540,1543,1544,1545,1546,1547,1548,1549,1552,1553,15
     54,1559,1560,1561,1562,1563,1564,1567,1568,1569,1570,1571,1572
730  DATA  1668,1669,1670,1671,1674,1677,1678,1679,1680,1689,1692,1693,16
     94,1695,1698
740  DATA  1802,1817,1826,1929,1930,1931,1944,1945,1946,1953,1954,1955,10
     73,1075,1088,1090,1097,1099
750  DATA  1201,1203,1216,1218,1225,1227,1329,1330,1331,1344,1345,1346,13
     53,1354,1355
760  DATA  1458,1473,1482,1586,1601,1610,1706,1707,1708,1709,1710,1711,17
     12,1713,1714,1715,1716,1717,1718,1719,1720,1721,1722
770  INPUT "ENTER VALUE FOR Z3 ";Z3$: CALL  - 998: CALL  - 868
780  DATA  1727,1728,1729,1730,1731,1732,1733,1734,1735,1736,1737,1738,17
     39,1740
790  DATA  99
800  HOME : VTAB 12: PRINT "GOODBYE ....."
```

Fig. 5-21 (cont). Listing for TEEPI program.

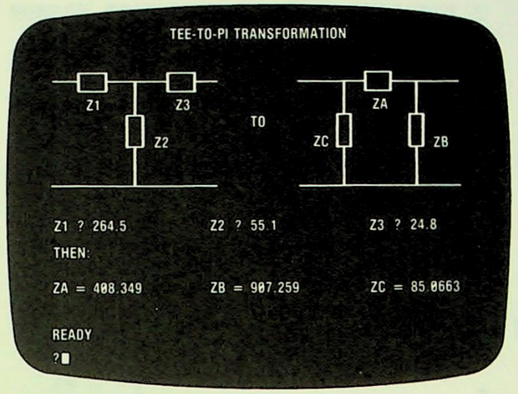


Fig. 5-22. Video results of Tee-to-Pi transformation.

Example 5-11

Determine the mesh currents I_a, I_b, and I_c for the network shown in Fig. 5-24, whose three mesh equations can be shown to be written as:

$$(30)I_a - (10)I_b - (20)I_c = 10$$
$$-(10)I_a + (90)I_b - (50)I_c = 0$$
$$-(20)I_a - (50)I_b + (110)I_c = 0$$

Fig. 5-25 shows the computed solutions for these equations so that I_a, I_b, and I_c are 0.477419, 0.135484, and 0.148387 amp respectively. It is then an easy manner to apply Ohm's law to determine the current through or the voltage across each resistor. For example, the current through the 20-ohm resistor is 0.477419 − 0.148387, or 0.329032 amp, while the voltage drop across it is (0.329032) (20), or 6.58 volts.

```
100  REM  SOLUTION OF SIMULTANEOUS EQUATIONS (SIMULEQU)
101  HOME : VTAB (7): PRINT  TAB( 14): INVERSE : PRINT "SOLUTION OF": NORMAL
102  PRINT : PRINT  TAB( 5): INVERSE : PRINT "SIMULTANEOUS LINEAR EQUATIO
     NS": NORMAL
103  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
104  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
105  FOR I = 1 TO 2500: NEXT I: CLEAR : DIM A(20,21),X(20): HOME
106  INPUT "NUMBER OF EQUATIONS OR VARIABLES? ";N
107  PRINT : INPUT "MINIMUM VALUE FOR ANY SOLUTION? ";MV
108  M = N + 1
109  FOR I = 1 TO N
110  PRINT : PRINT "COEFFICIENTS FOR EQUATION ";I;":"
111  FOR J = 1 TO M
112  PRINT "   #";J;
113  INPUT " ? ";A(I,J)
114  NEXT J
115  NEXT I
116  KK = 0:JJ = 0
117  FOR I = 1 TO N
118  JJ = KK + 1:LL = JJ:KK = KK + 1
119  IF ( ABS (A(JJ,KK)) - MV) <  = 0 THEN 121
120  GOTO 123
121  JJ = JJ + 1
122  GOTO 119
123  IF (LL - JJ) = 0 THEN 129
124  FOR MM = 1 TO M
125  Z = A(LL,MM)
126  A(LL,MM) = A(JJ,MM)
127  A(JJ,MM) = Z
128  NEXT MM
129  D = A(I,I)
130  FOR J = I TO M
131  A(I,J) = A(I,J) / D
132  NEXT J
133  K = I + 1
134  IF (K - M) < 0 THEN 136
```

Continued on next page.

Fig. 5-23. Listing for SIMULEQU program.

```
135  GOTO 143
136  FOR L = K TO N
137  Y = A(L,I)
138  FOR J = I TO M
139  A(L,J) = A(L,J) - A(I,J) * Y
140  NEXT J
141  NEXT L
142  NEXT I
143  X(N) = A(N,M)
144  L = N
145  FOR J = 2 TO N
146  S = 0
147  I = M + 1 - J
148  FOR K = I TO N
149  S = S + A(I - 1,K) * X(K)
150  NEXT K
151  L = L - 1
152  X(L) = A(I - 1,M) - S
153  NEXT J
154  PRINT : PRINT "THE SOLUTIONS ARE:"
155  FOR I = 1 TO N
156  PRINT "   X(";I;") = ";X(I)
157  NEXT I
158  END
```

Fig. 5-23 (cont). Listing for SIMULEQU program.

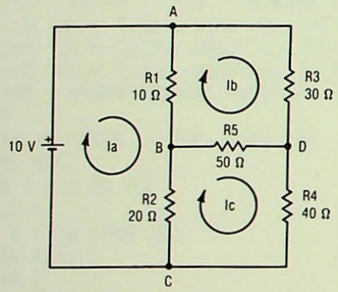


Fig. 5-24. Mesh currents network.

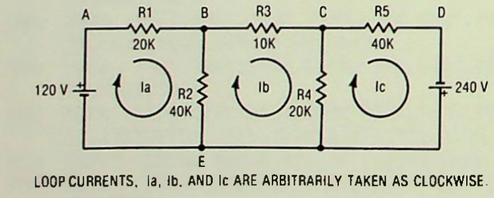


Fig. 5-26. Circuit for Example 5-12.

```
NUMBER OF EQUATIONS OR VARIABLES? 3

MINIMUM VALUE FOR ANY SOLUTION? 1E-6

COEFFICIENTS FOR EQUATION 1:
   #1 ? 30
   #2 ? -10
   #3 ? -20
   #4 ? 10

COEFFICIENTS FOR EQUATION 2:
   #1 ? -10
   #2 ? 90
   #3 ? -50
   #4 ? 0

COEFFICIENTS FOR EQUATION 3:
   #1 ? -20
   #2 ? -50
   #3 ? 110
   #4 ? 0


THE SOLUTIONS ARE:
   X(1) = .477419355
   X(2) = .135483871
   X(3) = .148387097
```

Fig. 5-25. Computed solutions for mesh currents.


```
NUMBER OF EQUATIONS OR VARIABLES? 5

MINIMUM VALUE FOR ANY SOLUTION? 1E-6

COEFFICIENTS FOR EQUATION 1:
   #1 ? 1
   #2 ? -1
   #3 ? 0
   #4 ? 0
   #5 ? 0
   #6 ? 0

COEFFICIENTS FOR EQUATION 2:
   #1 ? 0
   #2 ? 0
   #3 ? 1
   #4 ? -1
   #5 ? -1
   #6 ? 0

COEFFICIENTS FOR EQUATION 3:
   #1 ? 0
   #2 ? 0
   #3 ? 0
   #4 ? -20000
   #5 ? 40000
   #6 ? 240

COEFFICIENTS FOR EQUATION 4:
   #1 ? 0
   #2 ? -40000
   #3 ? 0
   #4 ? 20000
   #5 ? 0
   #6 ? 0

COEFFICIENTS FOR EQUATION 5:
   #1 ? 20000
   #2 ? 40000
   #3 ? 0
   #4 ? 0
   #5 ? 0
   #6 ? 120


THE SOLUTIONS ARE:
   X(1) = 4.90909091E-03
   X(2) = 5.45454547E-04
   X(3) = 4.36363637E-03
   X(4) = -1.09090909E-03
   X(5) = 5.45454545E-03
```

Fig. 5-27. Computed solutions for branch currents.

Example 5-12

Using the network of Fig. 5-26, determine the 5 branch currents using Kirchhoff's current and voltage laws.

At nodes A and B, application of the current law gives the two equations:

$$I_1 - I_2 - I_3 = 0$$

$$I_3 - I_4 - I_5 = 0$$

while the voltage law yields the following three equations:

$$-(20000)I_4 + (40000)I_5 = 240$$

$$-(40000)I_2 + (10000)I_3 + (20000)I_4 = 0$$

$$(20000)I_1 + (40000)I_2 = 120$$

so that there are 5 equations with 5 unknown currents. Fig. 5-27 shows the 5 computed solutions for this network.

The SIMULEQU program is not valid for complex coefficients, which result from circuits having capacitors and/or inductors in addition to resistors and voltage sources. However, simultaneous equations with complex coefficients are solved using the CPLXSEQ program listed in Fig. 5-28.

When run, the program requires the number of equations which is the same number of unknown variables and the minimum allowable value for any solution, which usually is set at 1E−6 (10^{-6}). The real and imaginary parts of each coefficient are then entered, one at a time for each equation, with the constant term entered last, as shown in the following example.

Example 5-13

For the circuit shown in Fig. 5-29, determine the three loop currents. Writing Kirchhoff's voltage law in rectangular form, we have:

$$(7+3j)I_1 - (5j)I_2 - (5)I_3 = 10/0 = 10$$

$$-(5j)I_1 + (12+3j)I_2 - (2-2j)I_3 = -5/30 = 4.33 - 2.5j$$

$$-(5)I_1 - (2-2j)I_2 + (17-2j)I_3 = -10/90 = -10j$$

Fig. 5-30 shows the computed results that were obtained.

```
100  REM  SIMULTANEOUS EQUATIONS WITH COMPLEX COEFFICIENTS (CPLXSEQ)
101  HOME : VTAB (7): PRINT  TAB( 9): INVERSE : PRINT "SIMUTANEOUS EQUATI
     ONS": NORMAL
102  PRINT : PRINT  TAB( 7): INVERSE : PRINT "WITH COMPLEX COEFFICIENTS":
     NORMAL
103  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
104  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
105  FOR I = 1 TO 2500: NEXT I
106  DIM A(10,22),B(20,22),P(10)
107  HOME : INPUT "NUMBER OF EQUATIONS? ";N
108  PRINT : INPUT "MINIMUM VALUE FOR ANY SOLUTION ? ";EP
109  MA = 2 * N - 1:MB = 2 * N + 1:MC = 2 * N:MD = 2 * N + 2
110  PRINT : PRINT "ENTER:": PRINT
111  FOR I = 1 TO N
112  PRINT "FOR EQUATION #";I
113  FOR J = 1 TO MD STEP 2
114  IF J = MD - 1 THEN 117
115  PRINT : PRINT "   COEFFICIENT "; INT (J / 2) + 1;": REAL PART ";
116  GOTO 118
117  PRINT : PRINT "   CONSTANT TERM: REAL PART ";
118  INPUT " ? ";A(I,J): PRINT  TAB( 19)"IMAG. PART ";
119  INPUT "? ";A(I,J + 1)
120  NEXT J
121  NEXT I
122  IK =  - 1
123  HOME : VTAB (9): PRINT  TAB( 8): FLASH : PRINT "COMPUTATIONS IN PROG
     RESS": NORMAL
124  FOR I = 1 TO MA STEP 2
125  IK = IK + 1
126  FOR J = 1 TO MB STEP 2
127  LJ = I - IK
128  B(I,J) = A(LJ,J):B(I,J + 1) =  - A(LJ,J + 1): NEXT J
129  NEXT I
130  FOR I = 2 TO MC STEP 2
131  FOR J = 1 TO MB STEP 2
132  LJ = I / 2:B(I,J) = A(LJ,J + 1):B(I,J + 1) = A(LJ,J): NEXT J
133  NEXT I
134  JJ = 1
135  FOR I = 1 TO MC
136  KK = I - 1
137  IF  ABS (B(I,I)) - EP <  > 0 THEN 144
138  KK = KK + 1:LL = 0
139  LL = LL + 1
140  FOR J = 1 TO MB
141  BT = B(KK,J):KL = KK + LL:B(KK,J) = B(KL,J)
142  B(KL,J) = BT: NEXT J
143  IF  ABS (B(I,I)) - EP = 0 THEN 139
144  BT = B(I,I)
145  FOR K = 1 TO MB:B(I,K) = B(I,K) / BT: NEXT K
146  JJ = JJ + 1
147  FOR J = JJ TO MC:BT = B(J,I)
148  FOR K = 1 TO MB:B(J,K) = B(J,K) - BT * B(I,K): NEXT K
149  NEXT J
150  NEXT I
151  B(MC,MB) = B(MC,MB) / B(MC,MC):B(MC,MC) = 1:JJ = 1:KK = 2
152  FOR I = 1 TO JJ
```

Continued on next page.

Fig. 5-28. Listing for CPLXSEQ program.


```
153  BT = B(I,KK)
154  FOR J = KK TO MB
155  B(I,J) = B(I,J) - BT * B(KK,J): NEXT J
156  NEXT I
157  JJ = JJ + 1:KK = KK + 1
158  IF JJ - MC <  > 0 THEN 152
159  K = 0
160  HOME : PRINT : PRINT "THE SOLUTIONS ARE:": PRINT
161  FOR I = 1 TO MC STEP 2
162  K = K + 1
163  PRINT "VALUE X(";K;"):"
164  PRINT : PRINT "REAL"; TAB( 10)"IMAGINARY"; TAB( 23)"MAGNITUDE"; TAB(
     34)"PHASE"
165  X(K) =  SQR ((B(I + 1,MB) ^ 2) + (B(I,MB) ^ 2))
166  B(I,MB) =  INT (B(I,MB) * 1E5 + .5) / 1E5:B(I + 1,MB) =  INT (B(I + 1
     ,MB) * 1E5 + .5) / 1E5:X(K) =  INT (X(K) * 1E5 + .5) / 1E5
167  PA =  ATN (B(I + 1,MB) / B(I,MB)) * 57.92577
168  IF B(I,MB) < 0 THEN 170
169  GOTO 171
170  P(K) = PA + 180: GOTO 172
171  P(K) = PA
172  P(K) =  INT (P(K) * 100 + .5) / 100
173  PRINT B(I,MB); TAB( 11);B(I + 1,MB); TAB( 24);X(K); TAB( 34);P(K)
174  PRINT : NEXT I
175  PRINT : END
```

Fig. 5-28 (cont). Listing for CPLXSEQ program.

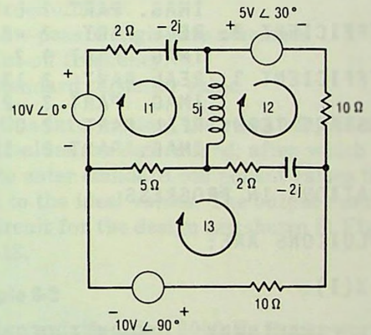


Fig. 5-29. Loop currents network.

```
NUMBER OF EQUATIONS? 3
MINIMUM VALUE FOR ANY SOLUTION? 1E-6
ENTER:
FOR EQUATION #1
   COEFFICIENT 1: REAL PART  ? 7
                  IMAG. PART ? 3
   COEFFICIENT 2: REAL PART  ? 0
                  IMAG. PART ? -5
   COEFFICIENT 3: REAL PART  ? 5
                  IMAG. PART ? 0
   CONSTANT TERM: REAL PART  ? 10
                  IMAG. PART ? 0

FOR EQUATION #2
   COEFFICIENT 1: REAL PART  ? 0
                  IMAG. PART ? -5
   COEFFICIENT 2: REAL PART  ? 12
                  IMAG. PART ? 3
   COEFFICIENT 3: REAL PART  ? -2
                  IMAG. PART ? 2
   CONSTANT TERM: REAL PART  ? -4.33
                  IMAG. PART ? -2.50

FOR EQUATION #3
   COEFFICIENT 1: REAL PART  ? -5
                  IMAG. PART ? 0
   COEFFICIENT 2: REAL PART  ? -2
                  IMAG. PART ? 2
   COEFFICIENT 3: REAL PART  ? 17
                  IMAG. PART ? -2
   CONSTANT TERM: REAL PART  ? 0
                  IMAG. PART ? -10

CALCULATIONS IN PROGRESS

THE SOLUTIONS ARE:

VALUE X(1):

REAL      IMAGINARY     MAGNITUDE    PHASE
.99953     -.22054       1.02357     -12.58

VALUE X(2):

REAL      IMAGINARY     MAGNITUDE    PHASE
-.27472    .12538        .30198      155.2

VALUE X(3):

REAL      IMAGINARY     MAGNITUDE    PHASE
.34296     -.56568       .66153      -59.42
```

Fig. 5-30. Computed results for loop currents.

6

Active Filter Design

This chapter describes programs for the rapid design of various types of active filters. Included are: the state-variable filter (3 op-amps); second- through sixth-order low-pass and high-pass filters having either Bessel (best delay), Butterworth (maximally flat), or Chebyshev (equal ripple) responses; bandpass filters; and notch filters. Each design determines the performance based on user selected standard resistor and capacitor values. For a more detailed discussion about active filters, either the *Design of Active Filters, With Experiments*, or the *Active-Filter Cookbook* (Howard W. Sams & Co., Inc., Indianapolis, IN 46268) are recommended references.

LOW-PASS AND HIGH-PASS FILTERS

The LPHP program listing shown in Fig. 6-1 enables us to design second- through sixth-order low-pass or high-pass filters having either Bessel, Butterworth, or Chebyshev (1-, 2-, or 3-dB) responses. Shown in Figs. 6-2 through 6-11 are the circuits of the low-pass and high-pass filters. Each second-order section is of the voltage-controlled voltage-source design (often referred to as a Sallen-Key filter) with both the frequency determining resistors R and capacitors C made equal. For the third- and fifth-order filters, the first section is a simple first-order passive network followed by a voltage follower. The damping values for each second-order section are controlled by the resistors RA and RB. Because the program restricts the filter's design to having both the frequency determining resistors and capacitors equal for each second-order section, the overall passband gain will be fixed, being set by the values chosen for the RA and RB resistors.

Example 6-1

Design a third-order, 350-Hz high-pass filter having a 1-dB Chebyshev response. When executed, the LPHP program requires the following initial information:

1. Type of filter.
2. Order of filter.
3. Low-pass or high-pass response.
4. Cut-off frequency.
5. Standard capacitor value.

From this information, the remaining ideal component values are determined, after which we are asked to enter standard component values that are closest to the ideal values. The output results and final circuit for the design are shown in Figs. 6-12 and 6-13.

Example 6-2

Design a sixth-order, 3000-Hz Butterworth low-pass filter. As shown in the output results in Fig. 6-14, we only have to determine the frequency determining resistors and capacitors for the first section, since they are the same for the remaining sections in a Butterworth design. The final circuit is shown in Fig. 6-15.

THE STATE-VARIABLE FILTER

The state-variable filter, shown in Fig. 6-16, uses three operational amplifiers, so that there are the following output responses: second-order low-pass, second-order high-pass, and a single-pole bandpass. Resistors R and capacitors C determine the cutoff (or center) frequency, while resistors RA and RB control the damping (or Q). For the

```
100  REM  LOW AND HIGH-PASS FILTER DESIGN (LPHP)
105  HOME : CLEAR : DIM F$(5),VF(120)
110  GOTO 1365
115  FOR V = 1 TO 110
120  READ VF(V): NEXT V
125  HOME : PRINT : PRINT F$(TY);" FILTER DESIGN"
130  PRINT : PRINT "SELECT:"
135  PRINT : PRINT "   LOW-PASS (1)": PRINT : INPUT "   HIGH-PASS (2) ";F
     R
140  PRINT : INPUT "ORDER OF FILTER ";O
145  IF O < 2 OR O > 6 THEN 140
150  PRINT : INPUT "CUTOFF FREQUENCY IN HZ ";FC
155  ON O - 1 GOTO 165,245,375,535,740
160  REM  2ND ORDER DESIGN
165  HOME : PRINT : PRINT "1ST SECTION (2ND ORDER):"
170  PRINT : INPUT "C IN UF ";C
175  K = VF(((TY - 1) * 22) + 1)
180  IF FR = 1 THEN 190
185  F = FC / K: GOTO 195
190  F = FC * K
195  R = 1000 / (6.28 * F * C): PRINT "   THEN R = "; INT (R * 100 + .5) /
     100;" K-OHMS"
200  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
205  IF Z$ = "Y" THEN 170
210  PRINT : INPUT "RA IN K-OHMS ";RA
215  D = VF((TY - 1) * 22 + 2)
220  RB = RA * (2 - D): PRINT : PRINT "   RB = "; INT (RB * 100 + .5) / 10
     0;" K-OHMS"
225  PRINT : INPUT "ANOTHER VALUE FOR RA (Y/N) ";Z$
230  IF Z$ = "Y" THEN 210
235  GOTO 975
240  REM  3RD ORDER DESIGN
245  HOME : PRINT : PRINT "1ST SECTION (1ST ORDER):"
250  PRINT : INPUT "C IN UF ";C1
255  K = VF(((TY - 1) * 22) + 3)
260  IF FR = 1 THEN 270
265  F1 = FC / K: GOTO 275
270  F1 = FC * K
275  R1 = 1000 / (6.28 * F1 * C1): PRINT : PRINT "   THEN R = "; INT (R1 *
     100 + .5) / 100;" K-OHMS"
280  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
285  IF Z$ = "Y" THEN 250
290  HOME : PRINT : PRINT "2ND SECTION (2ND ORDER):"
295  IF TY <  > 2 THEN 305
300  C2 = C1:R2 = R1: GOTO 345
305  PRINT : INPUT "C IN UF ";C2
310  K2 = VF(((TY - 1) * 22) + 5)
315  IF FR = 1 THEN 325
320  F2 = FC / K2: GOTO 330
325  F2 = FC * K2
330  R2 = 1000 / (6.28 * F2 * C2): PRINT : PRINT "   THEN R = "; INT (R2 *
     100 + .5) / 100;" K-OHMS"
335  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
340  IF Z$ = "Y" THEN 305
345  D = VF((TY - 1) * 22 + 6)
350  PRINT : INPUT "RA IN K-OHMS ";R3:R4 = R3 * (2 - D): PRINT : PRINT "
     THEN RB = "; INT (R4 * 100 + .5) / 100;" K-OHMS"
355  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
```

Continued on next page.

Fig. 6-1. Listing for LPHP program.


```
360  IF Z$ = "Y" THEN 350
365  GOTO 1040
370  REM  4TH ORDER DESIGN
375  HOME : PRINT : PRINT "1ST SECTION (2ND ORDER):"
380  K = VF(((TY - 1) * 22) + 7)
385  PRINT : INPUT "C IN UF ";C1
390  IF FR = 1 THEN 400
395  F1 = FC / K: GOTO 405
400  F1 = FC * K
405  R1 = 1000 / (6.28 * F1 * C1): PRINT : PRINT "   THEN R = "; INT (R1 *
     100 + .5) / 100;" K-OHMS"
410  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
415  IF Z$ = "Y" THEN 385
420  D = VF((TY - 1) * 22 + 8)
425  PRINT : INPUT "RA IN K-OHMS ";R3:R4 = R3 * (2 - D)
430  PRINT : PRINT "   THEN RB = "; INT (R4 * 100 + .5) / 100;" K-OHMS"
435  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
440  IF Z$ = "Y" THEN 425
445  HOME : PRINT : PRINT "2ND SECTION (2ND ORDER):"
450  IF TY <  > 2 THEN 460
455  C2 = C1:R2 = R1:F2 = F1: GOTO 500
460  PRINT : INPUT "C IN UF ";C2
465  K2 = VF(((TY - 1) * 22) + 9)
470  IF FR = 1 THEN 480
475  F2 = FC / K2: GOTO 485
480  F2 = FC * K2
485  R2 = 1000 / (6.28 * F2 * C2): PRINT : PRINT "   THEN R = "; INT (R2 *
     100 + .5) / 100;" K-OHMS"
490  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
495  IF Z$ = "Y" THEN 460
500  D2 = VF((TY - 1) * 22 + 10)
505  PRINT : INPUT "RA IN K-OHMS ";R5:R6 = R5 * (2 - D2)
510  PRINT : PRINT "   THEN RB = "; INT (R6 * 100 + .5) / 100;" K-OHMS"
515  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
520  IF Z$ = "Y" THEN 505
525  GOTO 1115
530  REM  5TH ORDER DESIGN
535  HOME : PRINT : PRINT "1ST SECTION (1ST ORDER):"
540  K = VF(((TY - 1) * 22) + 11)
545  PRINT : INPUT "C IN UF ";C1
550  IF FR = 1 THEN 560
555  F1 = FC / K: GOTO 565
560  F1 = FC * K
565  R1 = 1000 / (6.28 * F1 * C1): PRINT : PRINT "   THEN R = "; INT (R1 *
     100 + .5) / 100;" K-OHMS"
570  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
575  IF Z$ = "Y" THEN 545
580  HOME : PRINT : PRINT "2ND SECTION (2ND ORDER) :"
585  IF TY <  > 2 THEN 595
590  R2 = R1:C2 = C1:F2 = F1: GOTO 635
595  PRINT : INPUT "C IN UF ";C2
600  K2 = (((TY - 1) * 22) + 13)
605  IF FR = 1 THEN 615
610  F2 = FC / K2: GOTO 620
615  F2 = FC * K2
620  R2 = 1000 / (6.28 * F2 * C2): PRINT : PRINT "   THEN R = "; INT (R2 *
     100 + .5) / 100;" K-OHMS"
625  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
```

Continued on next page.

Fig. 6-1 (cont). Listing for LPHP program.

```
630  IF Z$ = "Y" THEN 595
635  D2 = VF((TY - 1) * 22 + 14)
640  PRINT : INPUT "RA IN K-OHMS ";R4:R6 = R4 * (2 - D2): PRINT : PRINT "
     THEN RB = "; INT (R6 * 100 + .5) / 100;" K-OHMS"
645  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
650  IF Z$ = "Y" THEN 640
655  HOME : PRINT : PRINT "3RD SECTION (2ND ORDER):"
660  IF TY < > 2 THEN 670
665  R3 = R1:C3 = C1:F3 = F1: GOTO 710
670  PRINT : INPUT "C IN UF ";C3
675  K3 = (((TY - 1) * 22) + 15)
680  IF FR = 1 THEN 690
685  F3 = FC / K3: GOTO 695
690  F3 = FC * K3
695  R3 = 1000 / (6.28 * F3 * C3): PRINT : PRINT "   THEN R = "; INT (R3 *
     100 + .5) / 100;" K-OHMS"
700  PRINT : INPUT "ANOTHER VALUE FOR C (Y/N) ? ";Z$
705  IF Z$ = "Y" THEN 670
710  D3 = VF((TY - 1) * 22 + 16)
715  PRINT : INPUT "RA IN K-OHMS";R5:R7 = R5 * (2 - D3): PRINT : PRINT "
     THEN RB = "; INT (R7 * 100 + .5) / 100;" K-OHMS"
720  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
725  IF Z$ = "Y" THEN 715
730  GOTO 1195
735  REM  6TH ORDER DESIGN
740  HOME : PRINT : PRINT "1ST SECTION (2ND ORDER):"
745  K = VF(((TY - 1) * 22) + 17)
750  PRINT : INPUT "C IN UF ";C1
755  IF FR = 1 THEN 765
760  F1 = FC / K: GOTO 770
765  F1 = FC * K
770  R1 = 1000 / (6.28 * F1 * C1): PRINT : PRINT "   THEN R = "; INT (R1 *
     100 + .5) / 100;" K-OHMS"
775  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
780  IF Z$ = "Y" THEN 750
785  D = VF((TY - 1) * 22 + 18)
790  PRINT : INPUT "RA IN K-OHMS ";R4:R7 = R4 * (2 - D)
795  PRINT : PRINT "   THEN RB = "; INT (R7 * 100 + .5) / 100;" K-OHMS"
800  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
805  IF Z$ = "Y" THEN 790
810  HOME : PRINT : PRINT "2ND SECTION (2ND ORDER):"
815  IF TY < > 2 THEN 825
820  R2 = R1:C2 = C1:F2 = F1: GOTO 865
825  PRINT : INPUT "C IN UF ";C2
830  K2 = VF(((TY - 1) * 22) + 19)
835  IF FR = 1 THEN 845
840  F2 = FC / K2: GOTO 850
845  F2 = FC * K2
850  R2 = 1000 / (6.28 * F2 * C2): PRINT : PRINT "   THEN R = "; INT (R2 *
     100 + .5) / 100;" K-OHMS"
855  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
860  IF Z$ = "Y" THEN 825
865  D2 = VF((TY - 1) * 22 + 20)
870  PRINT : INPUT "RA IN K-OHMS ";R5:R8 = R5 * (2 - D2)
875  PRINT : PRINT "   THEN RB = "; INT (R8 * 100 + .5) / 100;" K-OHMS"
880  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
885  IF Z$ = "Y" THEN 870
890  HOME : PRINT : PRINT "3RD SECTION (2ND ORDER):"
```

Continued on next page.

Fig. 6-1 (cont). Listing for LPHP program.


```
895  IF TY < > 2 THEN 905
900  F3 = F1:R3 = R1:C3 = C1: GOTO 945
905  PRINT : INPUT "C IN UF ";C3
910  K3 = VF(((TY - 1) * 22) + 21)
915  IF FR = 1 THEN 925
920  F3 = FC / K3: GOTO 930
925  F3 = FC * K3
930  R3 = 1000 / (6.28 * F3 * C3): PRINT : PRINT "   THEN R = "; INT (R3 *
     100 + .5) / 100;" K-OHMS"
935  PRINT : INPUT "TRY ANOTHER VALUE FOR C (Y/N) ";Z$
940  IF Z$ = "Y" THEN 905
945  D3 = VF((TY - 1) * 22 + 22)
950  PRINT : INPUT "RA IN K-OHMS ";R6:R9 = R6 * (2 - D3)
955  PRINT : PRINT "   THEN RB = "; INT (R9 * 100 + .5) / 100;" K-OHMS"
960  PRINT : INPUT "TRY ANOTHER VALUE FOR RA (Y/N) ";Z$
965  IF Z$ = "Y" THEN 950
970  GOTO 1275
975  HOME : PRINT "DESIGN SUMMARY:": PRINT "2ND ORDER ";F$(TY);" FILTER":
     PRINT
980  R =  INT (R * 100 + .5) / 100:RA =  INT (RA * 100 + .5) / 100:RB =  INT
     (RB * 100 + .5) / 100
985  PRINT : PRINT  TAB( 15)"1ST": PRINT  TAB( 13)"SECTION": PRINT
990  PRINT "C (UF)"; TAB( 13)C
995  PRINT "R (K-OHMS)"; TAB( 13)R
1000 PRINT "RA (K-OHMS)"; TAB( 13)RA
1005 PRINT "RB (K-OHMS)"; TAB( 13)RB: PRINT
1010 PRINT : PRINT "SELECT STANDARD VALUES FOR:": PRINT
1015 INPUT "RA (IN K-OHMS) ";RA
1020 INPUT "RB (IN K-OHMS) ";RB
1025 G = 1 + (RB / RA):DB = 20 *  LOG (G) /  LOG (10)
1030 PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
     " DB"
1035 GOTO 1495
1040 HOME : PRINT "DESIGN SUMMARY:": PRINT "3RD ORDER ";F$(TY);" FILTER"
     : PRINT
1045 R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100:R3 =
     INT (R3 * 100 + .5) / 100:R4 =  INT (R4 * 100 + .5) / 100
1050 PRINT : PRINT  TAB( 15)"1ST"; TAB( 25)"2ND"
1055 PRINT  TAB( 13)"SECTION"; TAB( 23)"SECTION": PRINT
1060 PRINT "C (UF)"; TAB( 13)C1; TAB( 23)C2
1065 PRINT "R (K-OHMS)"; TAB( 13)R1; TAB( 23)R2
1070 PRINT "RA (K-OHMS)"; TAB( 23)R3
1075 PRINT "RB (K-OHMS)"; TAB( 23)R4: PRINT
1080 PRINT
1085 PRINT : PRINT "SELECT STANDARD VALUES FOR:": PRINT
1090 INPUT "RA (IN K-OHMS) ";R3
1095 INPUT "RB (IN K-OHMS) ";R4
1100 G = 1 + (R4 / R3):DB = 20 *  LOG (G) /  LOG (10)
1105 PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
     " DB"
1110 GOTO 1495
1115 HOME : PRINT "DESIGN SUMMARY:": PRINT "4TH ORDER ";F$(TY);" FILTER:
     PRINT
1120 R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100:R3 =
     INT (R3 * 100 + .5) / 100
1125 R4 =  INT (R4 * 100 + .5) / 100:R5 =  INT (R5 * 100 + .5) / 100:R6 =
     INT (R6 * 100 + .5) / 100
1130 PRINT : PRINT  TAB( 15)"1ST"; TAB( 25)"2ND"
```

Continued on next page.

Fig. 6-1 (cont). Listing for LPHP program.

```
 1135  PRINT  TAB( 13)"SECTION"; TAB( 23)"SECTION": PRINT
 1140  PRINT "C (UF)"; TAB( 13)C1; TAB( 23)C2
 1145  PRINT "R (K-OHMS)"; TAB( 13)R1; TAB( 23)R2
 1150  PRINT "RA (K-OHMS)"; TAB( 13)R3; TAB( 23)R4
 1155  PRINT "RB (K-OHMS)"; TAB( 13)R5; TAB( 23)R6: PRINT
 1160  PRINT : PRINT "SELECT STANDARD VALUES FOR:"
 1165  PRINT
 1170  INPUT "RA(1)  (IN K-OHMS) ";R3: INPUT "RA(2)  (IN K-OHMS) ";R4
 1175  INPUT "RB(1)  (IN K-OHMS) ";R5: INPUT "RB(2)  (IN K-OHMS) ";R6
 1180  G = (1 + (R5 / R3)) * (1 + (R6 / R4)):DB = 20 *  LOG (G) /  LOG (10)
 1185  PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
       " DB"
 1190  GOTO 1495
 1195  HOME : PRINT "DESIGN SUMMARY:": PRINT "5TH ORDER ";F$(TY);" FILTER"
       : PRINT
 1200  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100:R3 =
       INT (R3 * 100 + .5) / 100:R4 =  INT (R4 * 100 + .5) / 100
 1205  R5 =  INT (R5 * 100 + .5) / 100:R6 =  INT (R6 * 100 + .5) / 100:R7 =
       INT (R7 * 100 + .5) / 100
 1210  PRINT : PRINT  TAB( 15)"1ST"; TAB( 25)"2ND"; TAB( 35)"3RD"
 1215  PRINT  TAB( 13)"SECTION"; TAB( 23)"SECTION"; TAB( 33)"SECTION": PRINT
 1220  PRINT "C (UF)"; TAB( 13)C1; TAB( 23)C2; TAB( 33)C3
 1225  PRINT "R (K-OHMS)"; TAB( 13)R1; TAB( 23)R2; TAB( 33)R3
 1230  PRINT "RA (K-OHMS)"; TAB( 23)R4; TAB( 33)R5
 1235  PRINT "RB (K-OHMS)"; TAB( 23)R6; TAB( 33)R7: PRINT
 1240  PRINT : PRINT "SELECT STANDARD VALUES FOR:"
 1245  PRINT
 1250  INPUT "RA(1)  (IN K-OHMS) ";R4: INPUT "RA(2)  (IN K-OHMS) ";R5
 1255  INPUT "RB(1)  (IN K-OHMS) ";R6: INPUT "RB(2)  (IN K-OHMS) ";R7
 1260  G = (1 + (R6 / R4)) * (1 + (R7 / R5)):DB = 20 *  LOG (G) /  LOG (10)
 1265  PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
       " DB"
 1270  GOTO 1495
 1275  HOME : PRINT "DESIGN SUMMARY:": PRINT "6TH ORDER ";F$(TY);" FILTER"
       : PRINT
 1280  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100:R3 =
       INT (R3 * 100 + .5) / 100
 1285  R4 =  INT (R4 * 100 + .5) / 100:R5 =  INT (R5 * 100 + .5) / 100:R6 =
       INT (R6 * 100 + .5) / 100
 1290  R7 =  INT (R7 * 100 + .5) / 100:R8 =  INT (R8 * 100 + .5) / 100:R9 =
       INT (R9 * 100 + .5) / 100
 1295  PRINT : PRINT  TAB( 15)"1ST"; TAB( 25)"2ND"; TAB( 35)"3RD"
 1300  PRINT  TAB( 13)"SECTION"; TAB( 23)"SECTION"; TAB( 33)"SECTION": PRINT
 1305  PRINT "C (UF)"; TAB( 13)C1; TAB( 23)C2; TAB( 33)C3
 1310  PRINT "R (K-OHMS)"; TAB( 13)R1; TAB( 23)R2; TAB( 33)R3
 1315  PRINT "RA (K-OHMS)"; TAB( 13)R4; TAB( 23)R5; TAB( 33)R6
 1320  PRINT "RB (K-OHMS)"; TAB( 13)R7; TAB( 23)R8; TAB( 33)R9: PRINT
 1325  PRINT : PRINT "SELECT STANDARD VALUES FOR:"
 1330  PRINT
 1335  INPUT "RA(1)  (IN K-OHMS) ";R4: INPUT "RA(2)  (IN K-OHMS) ";R5: INPUT
       "RA(3)  (IN K-OHMS) ";R6
 1340  INPUT "RB(1)  (IN K-OHMS) ";R7: INPUT "RB(2)  (IN K-OHMS) ";R8: INPUT
       "RB(3)  (IN K-OHMS) ";R9
 1345  G = (1 + (R7 / R4)) * (1 + (R8 / R5)) * (1 + (R9 / R6)):DB = 20 *  LOG
       (G) /  LOG (10)
 1350  PRINT : PRINT "PASSBAND GAIN FIXED AT "; INT (DB * 100 + .5) / 100;
       " DB"
 1355  GOTO 1495
 1360  REM   MENU SELECTION
```

Continued on next page.

Fig. 6-1 (cont). Listing for LPHP program.


```
 1365  HOME : VTAB (7): PRINT  TAB( 2): INVERSE : PRINT "LOW-PASS AND HIGH
       -PASS FILTER DESIGN": NORMAL
 1370  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)
       "HOWARD M. BERLIN, LTD."
 1375  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOF
       T VERSION"
 1380  FOR II = 1 TO 2500: NEXT II: HOME
 1385  PRINT : PRINT "SELECT TYPE OF FILTER RESPONSE:"
 1390  PRINT : PRINT "   1. BESSEL"
 1395  PRINT : PRINT "   2. BUTTERWORTH"
 1400  PRINT : PRINT "   3. 1-DB CHEBYSHEV"
 1405  PRINT : PRINT "   4. 2-DB CHEBYSHEV"
 1410  PRINT : PRINT "   5. 3-DB CHEBYSHEV"
 1415  PRINT : INPUT "CHOICE? ";TY
 1420  IF TY > 5 THEN 1365
 1425  F$(1) = "BESSEL":F$(2) = "BUTTERWORTH":F$(3) = "1-DB CHEBYSHEV"
 1430  F$(4) = "2-DB CHEBYSHEV":F$(5) = "3-DB CHEBYSHEV"
 1435  GOTO 115
 1440  REM  DESIGN PARAMETERS
 1445  DATA  1.274,1.732,1.328,0,1.454,1.447,1.436,1.916,1.610,1.241,1.557
       ,0
 1450  DATA  1.613,1.775,1.819,1.091,1.609,1.959,1.694,1.636,1.910,0.977
 1455  DATA  1,1.414,1,0,1,1,1,1.848,1,0.765,1,0
 1460  DATA  1,1.618,1,0.618,1,1.932,1,1.414,1,0.518
 1465  DATA  0.863,1.045,0.452,0,0.911,0.496,0.502,1.275,0.943,0.281,0.280
       ,0
 1470  DATA  0.634,0.714,0.961,0.181,0.347,1.314,0.733,0.455,0.977,0.125
 1475  DATA   0.852,0.895,0.322,0,0.913,0.412,0.466,1.088,0.946,0.234,0.22
       3,0
 1480  DATA  0.624,0.578,0.964,0.142,0.312,1.121,0.727,0.363,0.976,0.0989
 1485  DATA  0.841,0.767,0.299,0,0.916,0.326,0.443,0.929,0.950,0.179,0.178
       ,0
 1490  DATA  0.614,0.468,0.967,0.113,0.298,0.958,0.722,0.289,0.975,0.0782
 1495  CALL  - 756: END
```

Fig. 6-1 (cont). Listing for LPHP program.

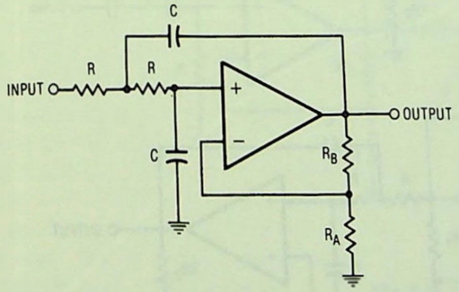


Fig. 6-2. Second-order low-pass filter.

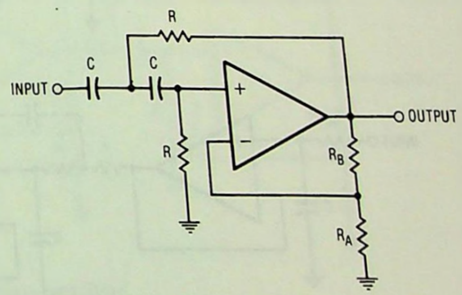


Fig. 6-3. Second-order high-pass filter.

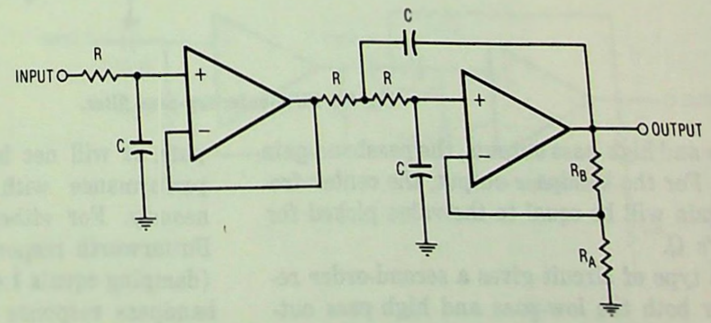


Fig. 6-4. Third-order low-pass filter.

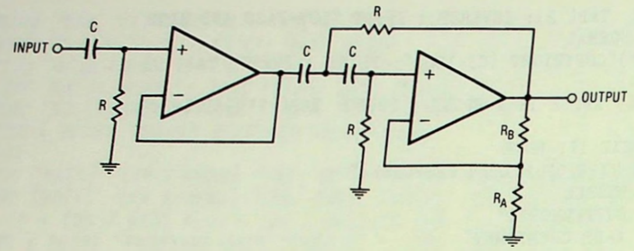


Fig. 6-5. Third-order high-pass filter.

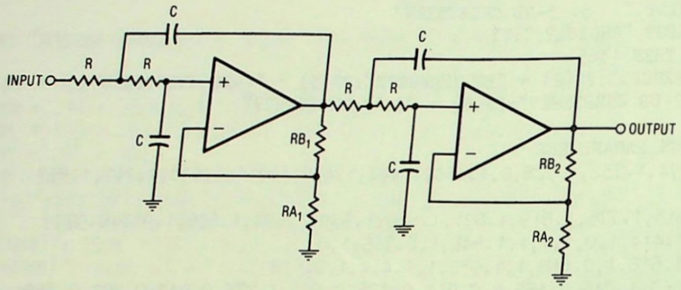


Fig. 6-6. Fourth-order low-pass filter.

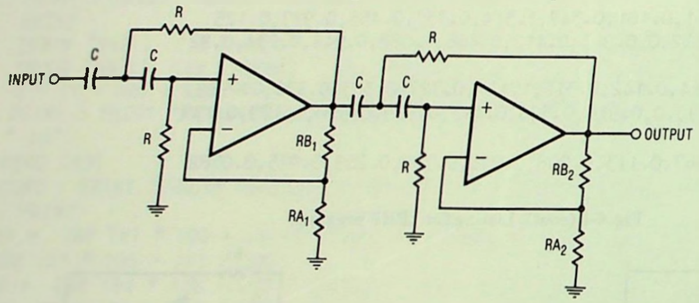


Fig. 6-7. Fourth-order high-pass filter.

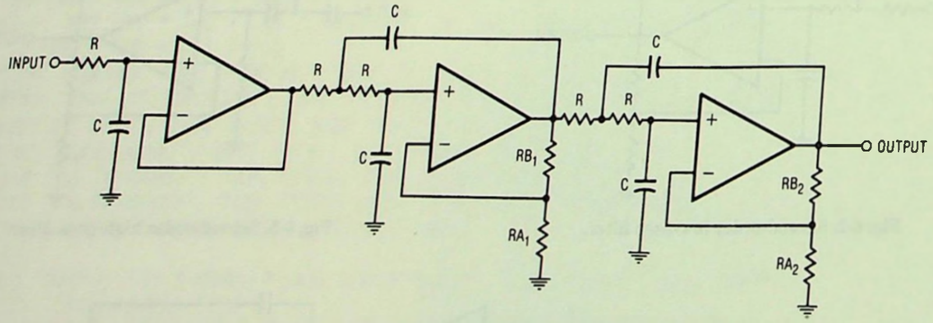


Fig. 6-8. Fifth-order low-pass filter.

low-pass and high-pass outputs, the passband gain is unity. For the bandpass output, the center frequency gain will be equal to the value picked for the filter's Q.

As this type of circuit gives a second-order response for both the low-pass and high-pass outputs, it will not be possible to obtain optimum performance with all three outputs simultaneously. For either a low-pass or a high-pass Butterworth response, Q must be equal to 0.707 (damping equals 1.414, or 1/Q). Consequently the bandpass response suffers terribly! Even for a

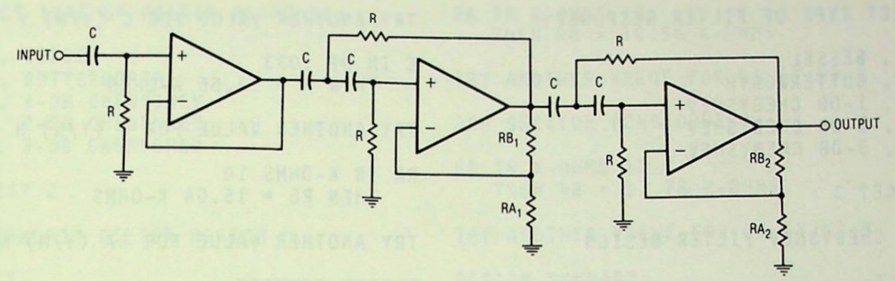


Fig. 6-9. Fifth-order high-pass filter.

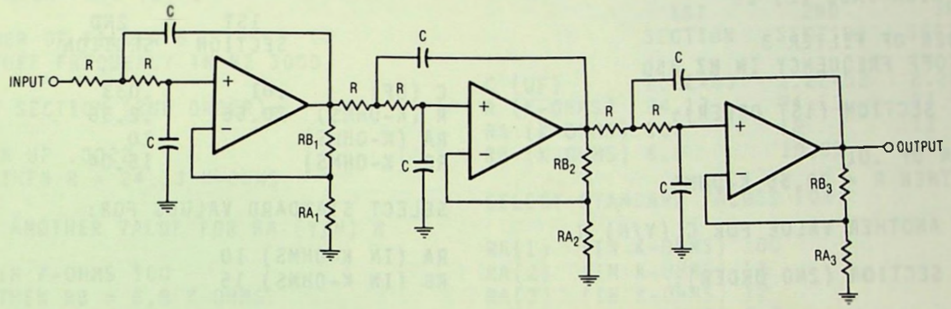


Fig. 6-10. Sixth-order low-pass filter.

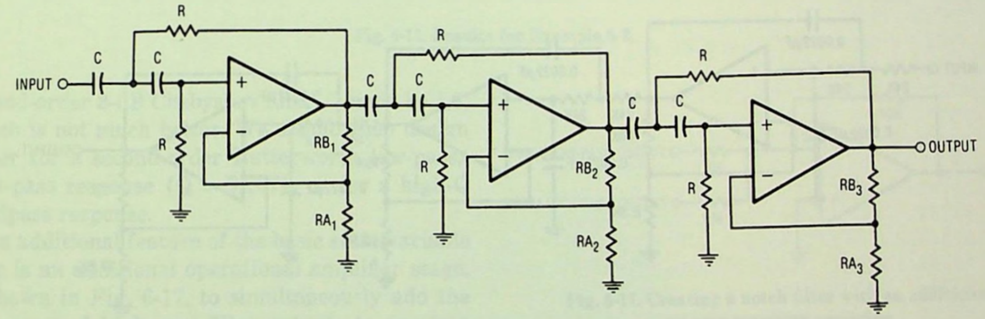


Fig. 6-11. Sixth-order high-pass filter.

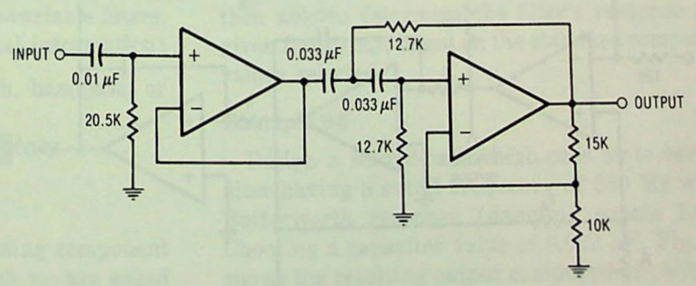


Fig. 6-13. Final circuit for Example 6-1.

```
SELECT TYPE OF FILTER RESPONSE:

   1. BESSEL
   2. BUTTERWORTH
   3. 1-DB CHEBYSHEV
   4. 2-DB CHEBYSHEV
   5. 3-DB CHEBYSHEV

CHOICE? 3

1-DB CHEBYSHEV FILTER DESIGN

SELECT:
   LOW-PASS (1)
   HIGH-PASS (2) 2

ORDER OF FILTER 3
CUTOFF FREQUENCY IN HZ 350

1ST SECTION (1ST ORDER):

C IN UF .01
   THEN R = 20.56 K-OHMS

TRY ANOTHER VALUE FOR C (Y/N) N

2ND SECTION (2ND ORDER):

C IN UF .022
   THEN R= 18.84 K-OHMS

TRY ANOTHER VALUE FOR C (Y/N) Y

C IN UF .033
   THEN R = 12.56 K-OHMS

TRY ANOTHER VALUE FOR C (Y/N) N

RA IN K-OHMS 10
   THEN RB = 15.04 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

DESIGN SUMMARY:
3RD ORDER 1-DB CHEBYSHEV FILTER

                  1ST       2ND
                SECTION   SECTION

C (UF)           .01       .033
R (K-OHMS)     20.56       12.56
RA (K-OHMS)                10
RB (K-OHMS)                15.04

SELECT STANDARD VALUES FOR:

RA (IN K-OHMS) 10
RB (IN K-OHMS) 15

PASSBAND GAIN FIXED AT 7.96 DB
```

Fig. 6-12. Output results for Example 6-1.

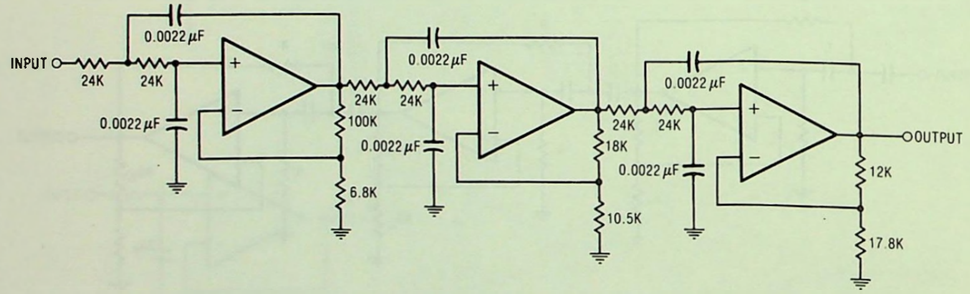


Fig. 6-15. Final circuit for Example 6-2.

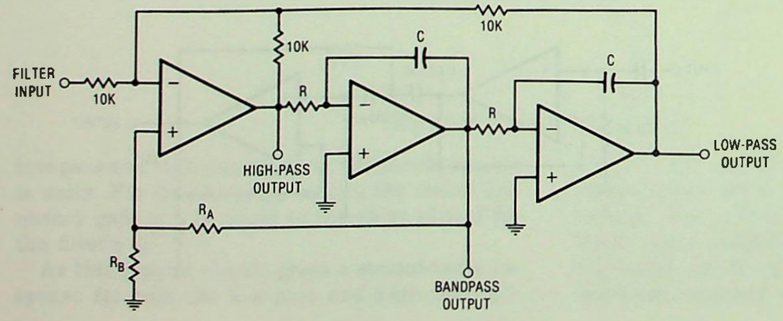


Fig. 6-16. Basic state-variable filter.


```
SELECT TYPE OF FILTER RESPONSE:

   1. BESSEL
   2. BUTTERWORTH
   3. 1-DB CHEBYSHEV
   4. 2-DB CHEBYSHEV
   5. 3-DB CHEBYSHEV

CHOICE? 2

BUTTERWORTH FILTER DESIGN

SELECT:
   LOW-PASS (1)
   HIGH-PASS (2) 1

ORDER OF FILTER 6
CUTOFF FREQUENCY IN HZ 3000

1ST SECTION (2ND ORDER):

C IN UF .0022
   THEN R = 24.13 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

RA IN K-OHMS 100
   THEN RB = 6.8 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

2ND SECTION (2ND ORDER):

RA IN K-OHMS 18
   THEN RB = 10.55 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

3RD SECTION (2ND ORDER):

RA IN K-OHMS 12
   THEN RB = 17.78 K-OHMS

TRY ANOTHER VALUE FOR RA (Y/N) N

DESIGN SUMMARY:
6TH ORDER BUTTERWORTH FILTER

                  1ST       2ND       3RD
                SECTION   SECTION   SECTION

C (UF)          2.2E-03   2.2E-03   2.2E-03
R (K-OHMS)        24.13     24.13     24.13
RA (K-OHMS)         100        18        12
RB (K-OHMS)         6.8     10.55     17.78

SELECT STANDARD VALUES FOR:

RA(1)  (IN K-OHMS) 100
RA(2)  (IN K-OHMS) 18
RA(3)  (IN K-OHMS) 12
RB(1)  (IN K-OHMS) 6.8
RB(2)  (IN K-OHMS) 10.5
RB(3)  (IN K-OHMS) 17.8

PASSBAND GAIN FIXED AT 12.46 DB
```

Fig. 6-14. Results for Example 6-2.

second-order 3-dB Chebyshev filter, Q must be 1.3, which is not much better. We should then design either for a second-order Butterworth low-pass/high-pass response (Q = 0.707), or for a high-Q bandpass response.

An additional feature of the basic state-variable filter is an additional operational amplifier stage, as shown in Fig. 6-17, to simultaneously add the low-pass and high-pass filter outputs to create a notch or band reject filter.

Shown in Fig. 6-18 is the program listing (SVFILTER) for designing state-variable filters, which requires the following initial information:

1. Type of response (low, high, bandpass, or notch).
2. Cutoff, center, or notch frequency.
3. Damping factor, or Q.
4. Standard capacitor value.

From this information, the remaining component values are determined, after which we are asked to enter standard component values that are closest to the ideal values. In addition, we are then able to determine the filter's response at a given frequency, based on the standard component values selected.

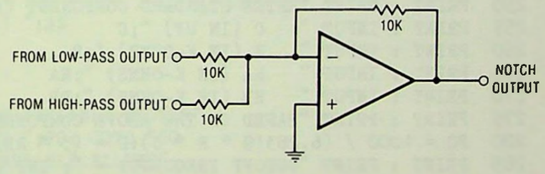


Fig. 6-17. Creating a notch filter with an additional op-amp summing amplifier.

Example 6-3

Design a second-order high-pass state-variable filter having a cutoff frequency of 350 Hz with a Butterworth response (damping equals 1.414). Choosing a capacitor value of 0.022 μF, Fig. 6-19 shows the resulting output computations, with the final circuit shown in Fig. 6-20.

```
100 REM  STATE-VARIABLE FILTER DESIGN (SVFILTER)
105 HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "STATE-VARIABLE FIL
    TER DESIGN": NORMAL
110 PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
    HOWARD M. BERLIN, LTD."
115 PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 12)"APPLESOFT
    VERSION"
120 FOR II = 1 TO 2500: NEXT II: HOME
125 PRINT "OUTPUT RESPONSES AVAILABLE:"
130 PRINT : PRINT "   1. 2ND ORDER LOW PASS."
135 PRINT : PRINT "   2. 2ND ORDER HIGH PASS."
140 PRINT : PRINT "   3. 1-POLE BANDPASS."
145 PRINT : PRINT "   4. 1-POLE NOTCH."
150 PRINT : INPUT "SELECT # ";Z9$
155 IF Z9$ <  > "1" AND Z9$ <  > "2" AND Z9$ <  > "3" AND Z9$ <  > "4" THEN
    150
160 IF Z9$ = "1" THEN 180
165 IF Z9$ = "2" THEN 355
170 IF Z9$ = "3" THEN 535
175 IF Z9$ = "4" THEN 535
180 HOME : PRINT : PRINT "2ND ORDER LOW-PASS SV FILTER"
185 PRINT : INPUT "ENTER CUTOFF FREQUENCY IN HZ ";FC
190 PRINT : INPUT "ENTER DAMPING FACTOR ";D
195 IF D < 3 THEN 205
200 INVERSE : PRINT : PRINT "DAMPING FACTOR TOO LARGE (>3).": PRINT : PRINT
    "PICK ANOTHER VALUE": NORMAL : GOTO 190
205 PRINT : PRINT "PASSBAND GAIN FIXED AT 1.0 (0 DB)"
210 PRINT : INPUT "ENTER CAPACITOR VALUE (IN UF) ";C:P = 6.28319 * FC
215 R = 1000 / (P * C):E = (3 / D) - 1
220 PRINT : PRINT "THEN...": PRINT "R = "; INT (R * 100 + .5) / 100;" K-
    OHMS"
225 PRINT : INPUT "ENTER VALUE FOR RA ";RA
230 RB = RA / E
235 PRINT "THEN...": PRINT "RB = "; INT (RB * 100 + .5) / 100;" K-OHMS"
240 PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";A$
245 IF A$ = "Y" THEN 210
250 PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR:"
255 PRINT : INPUT "   C (IN UF) ";C
260 PRINT : INPUT "   R (IN K-OHMS) ";R
265 PRINT : INPUT "   RA (IN K-OHMS) ";RA
270 PRINT : INPUT "   RB (IN K-OHMS) ";RB
275 PRINT : PRINT "BASED ON THE ABOVE COMPONENTS CHOSEN:"
280 FC = 1000 / (6.28319 * R * C):D = (3 * RB) / (RA + RB)
285 PRINT : PRINT "CUTOFF FREQUENCY = "; INT (FC * 10 + .5) / 10;" HZ"
290 PRINT : PRINT "DAMPING FACTOR = " INT (D * 1000 + .5) / 1000
295 PRINT : PRINT "TRY ANOTHER SET": INPUT "OF STANDARD VALUES (Y/N) ";A
    $
300 IF A$ = "Y" THEN 250
305 HOME : PRINT : INPUT "AMPLITUDE RESPONSE AT A GIVEN FREQUENCY (Y/N)
    ";A$
310 IF A$ = "N" THEN 350
315 PRINT : INPUT "FREQUENCY IN HZ ";FI
320 F = FI / FC
325 DB =  - 20 *  LOG ((F ^ 4) + ((D ^ 2 - 2) * F ^ 2) + 1) /  LOG (10)
330 DB = DB / 2
335 PRINT : PRINT "   AMPLITUDE RESPONSE = "; INT (DB * 100 + .5) / 100;
    " DB"
340 PRINT : INPUT "TRY ANOTHER FREQUENCY (Y/N)? ";A$
```

Continued on next page.

Fig. 6-18. Listing for SVFILTER program.


```
345 IF A$ = "Y" THEN 315
350 PRINT : PRINT "DESIGN COMPLETED": END
355 HOME : PRINT : PRINT "2ND ORDER HIGH-PASS SV FILTER": PRINT
360 PRINT : INPUT "ENTER CUTOFF FREQUENCY IN HZ ";FC
365 PRINT : INPUT "ENTER DAMPING FACTOR ";D
370 IF D < 3 THEN 385
375 INVERSE : PRINT : PRINT "DAMPING FACTOR TOO LARGE (>3).": PRINT "PIC
    K ANOTHER VALUE."
380 NORMAL : GOTO 365
385 PRINT : PRINT "PASSBAND GAIN FIXED AT 1.0 (0 DB)."
390 PRINT : INPUT "ENTER CAPACITOR VALUE IN UF ";C:P = 6.28319 * FC
395 R = 1000 / (P * C):E = (3 / D) - 1
400 PRINT "   THEN R = "; INT (R * 100 + .5) / 100;" K-OHMS"
405 PRINT : INPUT "ENTER VALUE FOR RA (IN K-OHMS) ";RA
410 RB = RA / E
415 PRINT : PRINT "   RB = "; INT (RB * 100 + .5) / 100;" K-OHMS"
420 PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";A$
425 IF A$ = "Y" THEN 390
430 PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR:"
435 PRINT : INPUT "   C (IN UF) ";C
440 PRINT : INPUT "   R (IN K-OHMS) ";R
445 PRINT : INPUT "   RA (IN K-OHMS) ";RA
450 PRINT : INPUT "   RB (IN K-OHMS) ";RB
455 PRINT : PRINT "BASED ON THE ABOVE COMPONENTES CHOSEN:"
460 FC = 1000 / (6.28319 * R * C):D = (3 * RB) / (RA + RB)
465 PRINT : PRINT "CUTOFF FREQUENCY = "; INT (FC * 10 + .5) / 10;" HZ"
470 PRINT : PRINT "DAMPING FACTOR = "; INT (D * 1000 + .5) / 1000
475 PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";A$
480 IF A$ = "Y" THEN 430
485 HOME : PRINT : PRINT "AMPLITUDE RESPONSE": INPUT "AT A GIVEN FREQUEN
    CY (Y/N) ";A$
490 IF A$ = "N" THEN 530
495 PRINT : INPUT "FREQUENCY IN HZ ";FI
500 F = FC / FI
505 DB =  - 20 *  LOG ((F ^ 4) + ((D ^ 2 - 2) * F ^ 2) + 1) /  LOG (10)
510 DB = DB / 2
515 PRINT "   AMPLITUDE RESPONSE = "; INT (DB * 100 + .5) / 100;" DB"
520 PRINT : INPUT "ADDITIONAL FREQUENCIES (Y/N) ";A$
525 IF A$ = "Y" THEN 495
530 PRINT : PRINT "DESIGN COMPLETED": END
535 IF Z9$ = "4" THEN 550
540 HOME : PRINT : PRINT "1-POLE BANDPASS STATE-VARIABLE FILTER"
545 PRINT : INPUT "ENTER CENTER FREQUENCY IN HZ ";FC: GOTO 560
550 HOME : PRINT : PRINT "1-POLE NOTCH STATE-VARIABLE FILTER"
555 PRINT : INPUT "ENTER NOTCH (NULL) FREQUENCY IN HZ ";FC
560 PRINT : INPUT "3 DB BANDWIDTH IN HZ ";BW:Q = FC / BW
565 PRINT : INPUT "ENTER CAPACITOR VALUE IN UF ";C
570 P = 6.28319 * FC
575 R = 1000 / (P * C)
580 E = (3 * Q) - 1
585 PRINT : PRINT "   THEN R = "; INT (R * 100 + .5) / 100;" K-OHMS"
590 PRINT : INPUT "ENTER VALUE FOR RA (IN K-OHMS) ";RA
595 RB = RA / E
600 PRINT : PRINT "   THEN RB = "; INT (RB * 100 + .5) / 100;" K-OHMS"
605 PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";A$
610 IF A$ = "Y" THEN 565
615 PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR:"
```

Continued on next page.

Fig. 6-18 (cont). Listing for SVFILTER program.

```
620  PRINT : INPUT "   C (IN UF) ";C
625  PRINT : INPUT "   R (IN K-OHMS) ";R
630  E = (3 * Q) - 1
635  PRINT : INPUT "   RA (IN K-OHMS) ";RA: PRINT : INPUT "   RB (IN K-OH
     MS) ";RB
640  IF Z9$ = "4" THEN 650
645  PRINT : PRINT "CENTER FREQUENCY GAIN FIXED AT Q ("; INT (Q * 10 + .5
     ) / 10;")": GOTO 655
650  PRINT : PRINT "PASSBAND GAIN FIXED AT 1.0 (0 DB)"
655  FC = 1000 / (6.28319 * R * C):Q = (1 + (RA / RB)) / 3:BW = FC / Q
660  FC =  INT (FC * 100 + .5) / 100:Q =  INT (Q * 10 + .5) / 10:BW =  INT
     (BW * 100 + .5) / 100
665  PRINT : PRINT "BASED ON ABOVE COMPONENT VALUES:"
670  IF Z9$ = "4" THEN 680
675  PRINT : PRINT "CENTER FREQUENCY = ";FC;" HZ": GOTO 685
680  PRINT : PRINT "NOTCH (NULL) FREQUENCY = ";FC;" HZ"
685  PRINT : PRINT "Q = ";Q
690  PRINT : PRINT "3 DB BANDWIDTH = ";BW;" HZ"
695  IF Z9$ = "4" THEN 710
700  DB = 8.68589 *  LOG (Q): PRINT : PRINT "CENTER FREQUENCY GAIN = "; INT
     (DB * 100 + .5) / 100;" DB"
705  GOTO 715
710  PRINT : PRINT "PASSBAND GAIN = 0 DB"
715  PRINT : PRINT "ANOTHER SET": INPUT "OF NOMINAL VALUES (Y/N) ";A$
720  IF A$ = "Y" THEN 615
725  HOME : PRINT : INPUT "AMPLITUDE RESPONSE AT A GIVEN FREQUENCY (Y/N)
     ";A$
730  IF A$ = "N" THEN 815
735  PRINT : INPUT "INPUT FREQUENCY IN HZ ";FI
740  F = FI / FC
745  IF Z9$ = "4" THEN 785
750  DB = 8.6859 *  LOG (Q) - 4.3429 *  LOG (1 + Q ^ 2 * ((F ^ 2 - 1) / F)
     ^ 2)
755  DB =  INT (DB * 100 + .5) / 100
760  PRINT : PRINT "   AMPLITUDE RESPONSE = ";DB;" DB"
765  PRINT : INPUT "ADDITIONAL FREQUENCIES (Y/N)? ";PP$
770  IF PP$ = "Y" THEN 735
775  IF PP$ = "N" THEN 815
780  GOTO 765
785  DB = 4.34294 *  LOG ((1 - F ^ 2) ^ 2) - 4.34294 *  LOG ((1 - F ^ 2) ^
     2 + (F / Q) ^ 2)
790  DB =  INT (DB * 100 + .5) / 100
795  PRINT : PRINT "   AMPLITUDE RESPONSE = ";DB;" DB"
800  PRINT : INPUT "ADDITIONAL FREQUENCIES (Y/N) ? ";PP$
805  IF PP$ = "Y" THEN 735
810  IF PP$ < > "N" THEN 800
815  PRINT : PRINT "DESIGN COMPLETED"
820  END
```

Fig. 6-18 (cont). Listing for SVFILTER program.

A BANDPASS FILTER FOR Q LESS THAN 10

For values of Q less than 10, the multiple-feedback circuit of Fig. 6-21 is often used, since it uses a minimum of components. Furthermore, the design is simplified by making both frequency determining capacitors equal. Resistors R1 and R3 determine the center frequency passband gain, while the center frequency is set by all three resistors. The program listing (BNDPASS1) is given in Fig. 6-22.

When executed the program requires the following initial information:

1. Design determined by knowing either Q, 3-dB bandwidth, or the 3-dB frequencies.
2. Standard capacitor value.


```
2ND ORDER HIGH-PASS SV FILTER

ENTER CUTOFF FREQUENCY IN HZ 350
ENTER DAMPING FACTOR 1.414
PASSBAND GAIN FIXED AT 1.0 (0 DB)

ENTER CAPACITOR VALUE IN UF .022
   THEN R = 20.67 K-OHMS

ENTER VALUE FOR RA (IN K-OHMS) 27
   RB = 24.07 K-OHMS

TRY ANOTHER SET OF VALUES (Y/N) N

ENTER STANDARD COMPONENT VALUES FOR:
   C (IN UF) .022
   R (IN K-OHMS) 20
   RA (IN K-OHMS) 27
   RB (IN K-OHMS) 24

BASED ON THE ABOVE COMPONENTS CHOSEN:

CUTOFF FREQUENCY = 361.7 HZ
DAMPING FACTOR = 1.412
TRY ANOTHER SET OF VALUES (Y/N) N

AMPLITUDE RESPONSE
AT A GIVEN FREQUENCY (Y/N) Y

FREQUENCY IN HZ 500
   AMPLITUDE RESPONSE = -1.04 DB

ADDITIONAL FREQUENCIES (Y/N) Y

FREQUENCY IN HZ 100
   AMPLITUDE RESPONSE = -22.36 DB

ADDITIONAL FREQUENCIES (Y/N) N

DESIGN COMPLETED
```

Fig. 6-19. Results for Example 6-3.

3. Center frequency.
4. Center frequency voltage gain.

after which the resistor values are determined. In addition, we are then able to determine the filter's response at any frequency based on the standard values chosen. The following example illustrates the BNDPASS1 program.

Example 6-4

Design a bandpass filter with a center frequency of 750 Hz, a voltage gain of 1.3 (+2.3 dB), and a Q of 4.2. Using a standard capacitor value of 0.01 μF, the three resistor values are calculated after which we are asked whether or not we want to try a different set of values, particularly that of the capacitor. If not, we are asked to enter standard component values that are closest to those calculated by the program. Again, we are asked whether or not we want to change the values. If not, then we are asked if we want to know the filter's frequency response at a particular frequency, based on the component values we have just chosen.

Figs. 6-23 and 6-24 show both the output and the final circuit of the program.

A BANDPASS FILTER FOR Q LESS THAN 50

The simple multiple-feedback circuit of Fig. 6-21 is not stable for values of Q greater than 10. However, relatively simple bandpass filters having Q values between 10 and 50 are possible with the addition of a second op-amp and three resistors, as shown in Fig. 6-25. Using the BNDPASS2 program of Fig. 6-26, the design of the filter is simplified by making both capacitors equal, as before. In addition, because of the nature of the design of the filter, the center frequency voltage gain must be greater than:

$$G > (Q)^{1/2} \qquad \text{(Eq. 6-1)}$$

so that high gain accompanies high Q.

Example 6-5

Design a bandpass filter having a 3-dB bandwidth of 50 Hz at a center frequency of 1000 Hz (so that **Q** = 20).

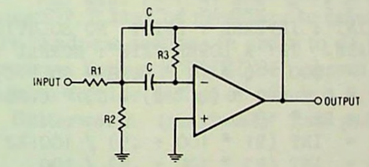


Fig. 6-21. Bandpass filter for Q less than 10.

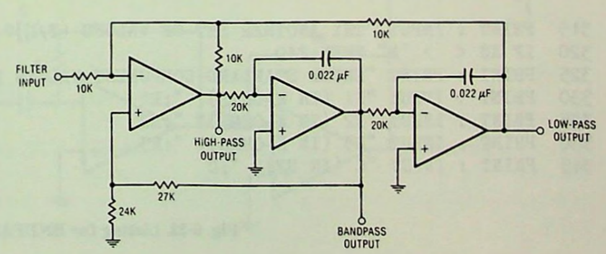


Fig. 6-20. Final circuit for Example 6-3.

```
100  REM    ** BANDPASS FILTER DESIGN (BNDPASS1) **
105  CLEAR : CALL  - 936: VTAB (5): PRINT  TAB( 9): INVERSE : PRINT "BAND
     PASS FILTER DESIGN": NORMAL : VTAB (7): PRINT  TAB( 11): INVERSE : PRINT
     "FOR Q LESS THAN 10": NORMAL
110  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980"
115  PRINT : PRINT  TAB( 9)"HOWARD M. BERLIN,LTD."
120  PRINT : PRINT  TAB( 16)"APPLE II"
125  PRINT : PRINT  TAB( 11)"APPLESOFT VERSION"
130  FOR I = 1 TO 2500: NEXT I: CALL  - 936
135  PRINT : PRINT "DESIGN BASED ON KNOWING:": PRINT : PRINT "     1. Q":
     PRINT : PRINT "     2. 3-DB BANDWIDTH": PRINT : PRINT "     3. UPPE
     R & LOWER 3-DB FREQUENCIES"
140  PRINT : INPUT "ENTER CHOICE ?";Z8: IF Z8 > 3 THEN 140
145  ON Z8 GOTO 150,170,180
150  PRINT : INPUT "Q ? ";Q
155  IF Q < 10 THEN 165
160  PRINT : INVERSE : PRINT "Q MUST BE LESS THAN 10": NORMAL : GOTO 150
165  GOTO 230
170  PRINT : INPUT "3-DB BANDWIDTH IN HZ? ";BW
175  GOTO 230
180  PRINT : INPUT "LOWER 3-DB FREQUENCY IN HZ? ";FL
185  PRINT : INPUT "UPPER 3-DB FREQUENCY IN HZ? ";FH
190  FO =  SQR (FH * FL)
195  PRINT : PRINT "    CENTER FREQUENCY = "; INT (FO * 10 + .5) / 10;" H
     Z"
200  BW = FH - FL:Q = FO / BW
205  PRINT : PRINT "    3-DB BANDWIDTH = "; INT (BW * 10 + .5) / 10;" HZ"
210  PRINT : PRINT "    Q = "; INT (Q * 10 + .5) / 10
215  IF Q <  = 10 THEN 225
220  PRINT : INVERSE : PRINT "Q MUST BE LESS THAN 10.": PRINT "INCREASE B
     ANDWIDTH": NORMAL : GOTO 180
225  GOTO 230
230  PRINT : PRINT "HIT RETURN TO CONTINUE";
235  CALL  - 756
240  CALL  - 936: PRINT : INPUT "STANDARD CAPACITOR VALUE IN UF? ";C
245  ON Z8 GOTO 250,250,265
250  PRINT : INPUT "CENTER FREQUENCY IN HZ? ";FO
255  IF Z8 = 1 THEN 265
260  Q = FO / BW
265  PRINT : INPUT "CENTER FREQUENCY VOLTAGE GAIN? ";G
270  IF Q >  =  SQR (G / 2) THEN 280
275  PRINT : INVERSE : PRINT "NO SOLUTION POSSIBLE FOR STATED Q AND": PRINT
     "GAIN.  TRY A LOWER GAIN": NORMAL : GOTO 265
280  R1 = 1000 * Q / (G * 6.28 * FO * C):R3 = 2000 * Q / (6.28 * FO * C)
285  R2 = Q / ((2 * (Q ^ 2) - G) * 6.28 * FO * C)
290  R2 = R2 * 1000
295  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100
300  R3 =  INT (R3 * 100 + .5) / 100
305  PRINT : PRINT "THEN:": PRINT " R1"," R2"," R3"
310  PRINT : PRINT : PRINT R1,R2,R3: PRINT : PRINT "(ALL VALUES IN K-OHMS
     )"
315  PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N)? ";X$
320  IF X$ <  > "N" THEN 240
325  PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR: "
330  PRINT : INPUT "R1 (IN K-OHMS)? ";R1
335  PRINT : INPUT "R2 (IN K-OHMS)? ";R2
340  PRINT : INPUT "R3 (IN K-OHMS)? ";R3
345  PRINT : INPUT "C (IN UF)? ";C
```

Continued on next page.

Fig. 6-22. Listing for BNDPASS1 program.


```
350  PRINT : PRINT : PRINT "BASED ON THE ABOVE STANDARD COMPONENTS": PRINT
     "CHOSEN:"
355  R3 = 1E3 * R3:R2 = 1E3 * R2:R1 = 1E3 * R1:C = 1E - 6 * C:Y = 1 / (R3 *
     C ^ 2)
360  R1 = 1 / R1:R2 = 1 / R2:FO =  SQR (Y * (R1 + R2)):FO = FO / 6.28
365  G = R3 * R1 / 2
370  Q = G * 6.28 * FO * C / R1
375  PRINT : PRINT "CENTER FREQUENCY = "; INT (FO * 10 + .5) / 10;" HZ"
380  PRINT : PRINT "CENTER FREQUENCY VOLTAGE GAIN = "; INT (G * 100 + .5)
      / 100
385  PRINT : PRINT "Q = "; INT (Q * 100 + .5) / 100,"BANDWIDTH = "; INT (
     (FO / Q) * 100 + .5) / 100;" HZ"
390  PRINT : PRINT "TRY ANOTHER SET OF STANDARD": INPUT "VALUES (Y/N)? ";
     X$
395  IF X$ <  > "N" THEN 330
400  HOME : PRINT : PRINT "DO YOU WISH TO KNOW THE AMPLITUDE": INPUT "RES
     PONSE (Y/N)? ";X$
405  IF X$ <  > "Y" THEN 440
410  PRINT : INPUT "ENTER INPUT FREQUENCY IN HZ? ";FI
415  F = FI / FO:FX = (((F ^ 2) - 1) / F) ^ 2:V = 1 + (Q ^ 2) * FX
420  DB = (20 *  LOG (G) /  LOG (10)) - 10 *  LOG (V) /  LOG (10):DB =  INT
     (DB * 100 + .5) / 100
425  PRINT : PRINT "     AMPLITUDE RESPONSE = ";DB;" DB"
430  PRINT : INPUT "TRY ANOTHER FREQUENCY (Y/N)? ";X$
435  IF X$ = "Y" THEN 410
440  PRINT : PRINT "DESIGN COMPLETED"
445  END
```

Fig. 6-22 (cont). Listing for BNDPASS1 program.

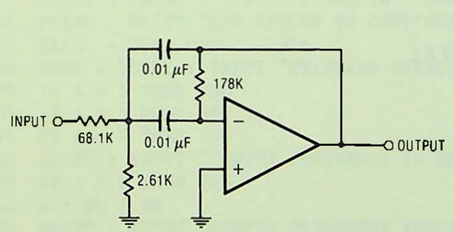
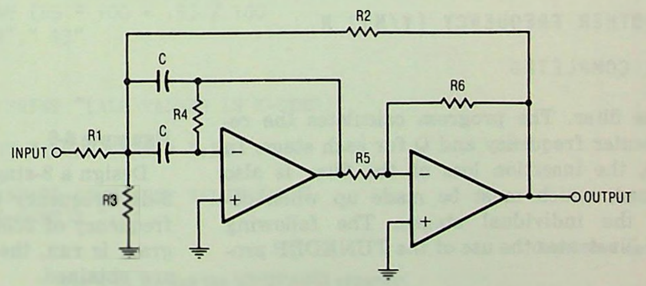


Fig. 6-24. Final circuit for Example 6-4.

Using a standard value of 0.018 μF and a center frequency voltage gain of 5, the output results are shown in Fig. 6-27—the final circuit in Fig. 6-28.

Fig. 6-25. Bandpass filter for Q less than 50.



STAGGERED-TUNED BUTTERWORTH BANDPASS FILTERS

Very often it is required to have a bandpass filter that passes a range of frequencies, instead of a single frequency. In most cases a single stage bandpass section like those previously described will not offer the necessary response. Consequently, it may be necessary to cascade two or more bandpass sections in order to do the job. The program listing (TUNEDBP) shown in Fig. 6-29 allows for the design of either a 2, 3, or 4 stage Butterworth (maximally flat passband)

```
DESIGN BASED ON KNOWING:
     1. Q
     2. 3-DB BANDWIDTH
     3. UPPER & LOWER 3-DB FREQUENCIES

ENTER CHOICE ?1

Q ? 4.2

HIT RETURN TO CONTINUE
STANDARD CAPACITOR VALUE IN UF? .01
CENTER FREQUENCY IN HZ? 750
CENTER FREQUENCY VOLTAGE GAIN? 1.3

THEN:
 R1              R2              R3

68.59           2.62            178.34

(ALL VALUES IN K-OHMS)

TRY ANOTHER SET OF VALUES (Y/N)? N
ENTER STANDARD COMPONENT VALUES FOR:

R1 (IN K-OHMS)? 68.1
R2 (IN K-OHMS)? 2.61
R3 (IN K-OHMS)? 178
C (IN UF)? .01


BASED ON THE ABOVE STANDARD COMPONENTS CHOSEN:
CENTER FREQUENCY = 752.8 HZ
CENTER FREQUENCY VOLTAGE GAIN = 1.31
Q = 4.21        BANDWIDTH = 178.92 HZ

TRY ANOTHER SET OF STANDARD
VALUES (Y/N)? N

DO YOU WISH TO KNOW THE AMPLITUDE
RESPONSE (Y/N)? Y
ENTER INPUT FREQUENCY IN HZ? 400
     AMPLITUDE RESPONSE = -12.9 DB

TRY ANOTHER FREQUENCY (Y/N)? Y

ENTER INPUT FREQUENCY IN HZ? 1125
     AMPLITUDE RESPONSE = -8.83 DB

TRY ANOTHER FREQUENCY (Y/N)? N

DESIGN COMPLETED
```

Fig. 6-23. Results for Example 6-4.

bandpass filter. The program calculates the required center frequency and Q for each stage. In addition, the insertion loss of the filter is also determined, which must be made up when designing the individual stages. The following example illustrates the use of the TUNEDBP program.

Example 6-6

Design a 3-stage bandpass filter having a lower 3-dB frequency of 2050 Hz and an upper 3-dB frequency of 2350 Hz. When the TUNEDBP program is run, the results, as shown in Fig. 6-30, are obtained.


```
100  REM  BANDPASS FILTER DESIGN FOR Q<50 (BNDPASS2)
105  HOME : VTAB (7): PRINT  TAB( 9): INVERSE : PRINT "BANDPASS FILTER DE
     SIGN": NORMAL
110  PRINT : PRINT  TAB( 11): INVERSE : PRINT "FOR Q LESS THAN 50": NORMAL
115  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD. "
120  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
125  FOR I = 1 TO 2500: NEXT I: HOME
130  HOME : PRINT "DESIGN BASED ON KNOWING:": PRINT : PRINT : PRINT  TAB(
     5)"1. Q"
135  PRINT : PRINT  TAB( 5)"2. 3-DB BANDWIDTH"
140  PRINT : PRINT  TAB( 5)"3. UPPER & LOWER 3-DB FREQUENCIES"
145  PRINT : INPUT "ENTER CHOICE? ";A
150  ON A GOTO 155,180,190
155  PRINT : INPUT "Q ? ";Q
160  IF Q >  = 50 THEN 170
165  GOTO 175
170  PRINT : INVERSE : PRINT "Q MUST BE LESS THAN 50.": NORMAL : GOTO 155
175  GOTO 220
180  PRINT : INPUT "3-DB BANDWIDTH IN HZ? ";BW
185  GOTO 220
190  PRINT : INPUT "LOWER 3-DB FREQUENCY IN HZ? ";FL
195  PRINT : INPUT "UPPER 3-DB FREQUENCY IN HZ? ";FH
200  FO =  SQR (FH * FL): PRINT : PRINT "   CENTER FREQUENCY = "; INT (FO *
     10 + .5) / 10;" HZ"
205  BW = FH - FL:Q = FO / BW
210  PRINT : PRINT "   3-DB BANDWIDTH = " INT (BW * 10 + .5) / 10;" HZ"
215  PRINT : PRINT "   Q = "; INT (Q * 100 + .5) / 100
220  PRINT : PRINT "HIT RETURN TO CONTINUE";
225  CALL  - 756: CALL  - 936
230  HOME : PRINT : INPUT "STANDARD CAPACITOR VALUE IN UF? ";C
235  IF A = 1 THEN 250
240  IF A = 3 THEN 265
245  IF A = 2 THEN 250
250  PRINT : INPUT "CENTER FREQUENCY IN HZ? ";FO
255  IF A = 1 THEN 265
260  Q = FO / BW
265  PRINT : INPUT "CENTER FREQUENCY VOLTAGE GAIN? ";G
270  IF G >  SQR (Q) THEN 280
275  PRINT : INVERSE : PRINT "GAIN TOO LOW.  TRY ANOTHER VALUE": NORMAL :
     GOTO 265
280  K = G /  SQR (Q):R1 = Q * 1000 / (6.28 * FO * C)
285  R2 = R1 * K * Q / ((2 * Q) - 1)
290  R6 = K * R1
295  DM = (Q ^ 2) - 1 - (2 / K) + 1 / (K * Q):R3 = R1 / DM
300  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100
305  R3 =  INT (R3 * 100 + .5) / 100:R4 =  INT (R4 * 100 + .5) / 100
310  R5 = R1:R4 = R1:R6 =  INT (R6 * 100 + .5) / 100
315  PRINT : PRINT " R1"," R2"," R3"
320  PRINT R1,R2,R3: PRINT
325  PRINT " R4"," R5"," R6"
330  PRINT R4,R5,R6: PRINT : PRINT "(ALL VALUES IN K-OHMS)"
335  PRINT
340  INPUT "TRY ANOTHER SET OF VALUES (Y/N)? ";X$
345  IF X$ = "Y" THEN 230
350  PRINT : PRINT "ENTER STANDARD COMPONENT VALUE FOR:"
355  PRINT : INPUT " R1 (IN K-OHMS)? ";R1
```

Continued on next page.

Fig. 6-26. Listing for BNDPASS2 program.

```
360  INPUT " R2 (IN K-OHMS)? ";R2
365  INPUT " R3 (IN K-OHMS)? ";R3
370  INPUT " R6 (IN K-OHMS)? ";R6
375  INPUT " C (IN UF)? ";C
380  PRINT : PRINT "BASED ON ABOVE STANDARD COMPONENTS": PRINT "CHOSEN:"
385  K = R6 / R1:XX = (1000 / (6.28 * C * R1)) *  SQR ((R1 + 2 * R3) / R3)
390  FO = XX:Q = 6.28 * R1 * FO * C / 1000
395  G = K *  SQR (Q)
400  PRINT : PRINT "CENTER FREQUENCY = "; INT (FO * 10 + .5) / 10;" HZ"
405  PRINT : PRINT "CENTER FREQUENCY GAIN = "; INT (G * 100 + .5) / 100
410  PRINT : PRINT "Q = "; INT (Q * 100 + .5) / 100
415  PRINT : PRINT "TRY ANOTHER SET OF STANDARD": INPUT "VALUES (Y/N)? ";
     X$
420  IF X$ = "Y" THEN 350
425  HOME : PRINT "DO YOU WISH TO KNOW THE AMPLITUDE": INPUT "RESPONSE (Y
     /N)? ";X$
430  IF X$ = "Y" THEN 440
435  GOTO 470
440  PRINT : INPUT "INPUT FREQUENCY IN HZ? ";FI
445  F = FI / FO:FX = (((F ^ 2) - 1) / F) ^ 2:V = 1 + (Q ^ 2) * FX
450  DB = (20 *  LOG (G) /  LOG (10)) - (10 *  LOG (V) /  LOG (10))
455  PRINT : PRINT "   AMPLITUDE  RESPONSE = "; INT (DB * 100 + .5) / 100
     ;" DB"
460  PRINT : INPUT "TRY ANOTHER FREQUENCY (Y/N)? ";X$
465  IF X$ = "Y" THEN 440
470  PRINT : PRINT "DESIGN COMPLETED"
475  END
```

Fig. 6-26 (cont). Listing for BNDPASS2 program.

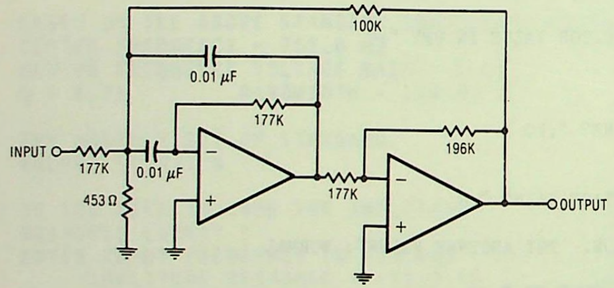


Fig. 6-28. Final circuit for Example 6-5.

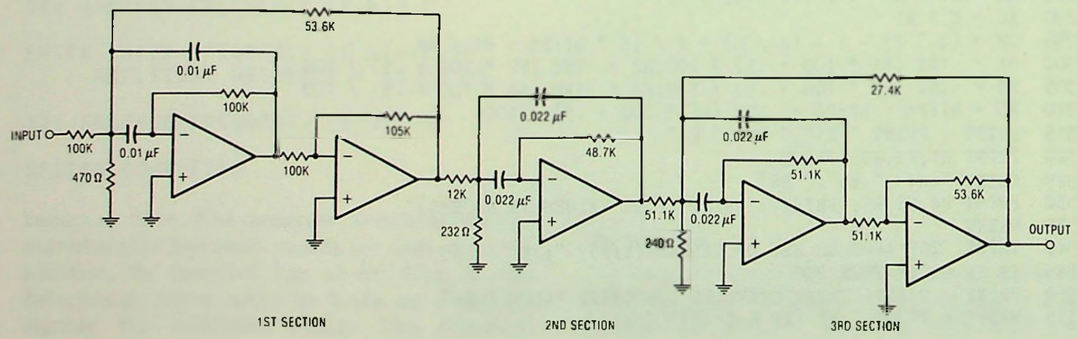


Fig. 6-31. Final circuit for Example 6-6.


```
        BANDPASS FILTER DESIGN FOR Q LESS THAN 50
DESIGN BASED ON KNOWING:
     1. Q
     2. 3-DB BANDWIDTH
     3. UPPER & LOWER 3-DB FREQUENCIES

ENTER CHOICE ?2

3-DB BANDWIDTH IN HZ? 50

HIT RETURN TO CONTINUE
STANDARD CAPACITOR VALUE IN UF? .018
CENTER FREQUENCY IN HZ? 1000

CENTER FREQUENCY VOLTAGE GAIN? 3

GAIN TOO LOW.  TRY ANOTHER VALUE

CENTER FREQUENCY VOLTAGE GAIN? 5

THEN:
 R1              R2              R3
176.93          101.44           .45

 R4              R5              R6
176.93          176.93          197.81

(ALL VALUES IN K-OHMS)

TRY ANOTHER SET OF VALUES (Y/N)? N
ENTER STANDARD COMPONENT VALUE FOR:

 R1 (IN K-OHMS)? 177
 R2 (IN K-OHMS)? 100
 R3 (IN K-OHMS)? 0.453
 R6 (IN K-OHMS)? 196
 C (IN UF)? .018

BASED ON THE ABOVE STANDARD COMPONENTS CHOSEN:
CENTER FREQUENCY = 990.5 HZ
CENTER FREQUENCY GAIN = 4.93
Q = 19.82

TRY ANOTHER SET OF STANDARD
VALUES (Y/N)? N

DO YOU WISH TO KNOW THE AMPLITUDE
RESPONSE (Y/N)? Y

INPUT FREQUENCY IN HZ? 990.5
     AMPLITUDE REPONSE = 13.86 DB

TRY ANOTHER FREQUENCY (Y/N)? Y

INPUT FREQUENCY IN HZ? 300
     AMPLITUDE REPONSE = -21.62 DB
```

Continued on next page.

Fig. 6-27. Results for Example 6-5.

```
TRY ANOTHER FREQUENCY (Y/N)? N

DESIGN COMPLETED
```

Fig. 6-27 (cont). Results for Example 6-5.

```
 STAGGER TUNED BUTTERWORTH BANDPASS FILTER DESIGN
NUMBER OF STAGES DESIRED? 3
LOWER 3-DB FREQUENCY (IN HZ)? 2050
UPPER 3-DB FREQUENCY (IN HZ)? 2350
   CENTER FREQUENCY = 2194.9 HZ
   3-DB BANDWIDTH = 300 HZ

1ST SECTION DESIGN PARAMETERS:
 CENTER FREQUENCY = 2329.9 HZ   Q = 14.6

2ND SECTION DESIGN PARAMETERS:
 CENTER FREQUENCY = 2194.9 HZ   Q = 7.3

3RD SECTION DESIGN PARAMETERS:
 CENTER FREQUENCY = 2067.7 HZ   Q = 14.6

GAIN OF CASCADED STAGES MUST BE AT
LEAST 12.2 DB

AMPLITUDE RESPONSE (Y/N)? Y
INPUT FREQUENCY IN HZ? 1750

   AMPLITUDE RESPONSE WITH INSERTION
   LOSS = -43.5 DB

ANOTHER INPUT FREQUENCY (Y/N)? N

DESIGN COMPLETED
```

Fig. 6-30. Results for Example 6-6.

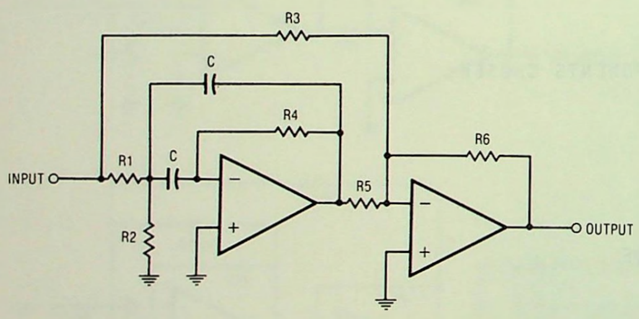


Fig. 6-32. Notch filter for Q less than 25.

For our example, we see that the first and third sections have values of Q that are greater than 10 (i.e., 14.6). Consequently, we should use the filter circuit shown in Fig. 6-25 and the BNDPASS2 program to calculate each section separately. For the second section, the Q is less than 10 (i.e., 7.3), so that we are able to use the circuit of Fig. 6-21 and the corresponding BNDPASS1 program. If the Q of any stage is found to be greater than 50, then the state-variable filter should be used. In addition, the overall gain of the three stages must be at least 12.16 dB, or a voltage gain of 4.05 in order to overcome the insertion loss presented by the filter.

Fig. 6-31 shows one of many circuits that are possible to satisfy our design. The voltage gain


```
100  REM  STAGGERED-TUNED BUTTERWORTH BANDPASS FILTER DESIGN (TUNEDBP)
101  CLEAR : HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "STAGGERED
     TUNED BUTTERWORTH": NORMAL
102  PRINT : PRINT  TAB( 9): INVERSE : PRINT "BANDPASS FILTER DESIGN": NORMAL
     : PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980"
103  PRINT : PRINT  TAB( 9)"HOWARD M. BERLIN,LTD.": PRINT : PRINT  TAB( 1
     6)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT VERSION"
104  FOR I = 1 TO 2500: NEXT I: HOME
105  INPUT "NUMBER OF STAGES DESIRED? ";P: IF P > 4 GOTO 187
106  PRINT : INPUT "LOWER 3-DB FREQUENCY (IN HZ)? ";FL
107  PRINT : INPUT "UPPER 3-DB FREQUENCY (IN HZ)? ";FH
108  FZ = SQR (FL * FH): PRINT "   CENTER FREQUENCY = "; INT (FZ * 10 + .
     5) / 10;" HZ"
109  B = FH - FL: PRINT "   3-DB BANDWIDTH = "; INT (B * 10 + .5) / 10;" H
     Z"
110  B1 = B / FZ
111  IF P > 2 THEN 134
112  PRINT : PRINT "1ST SECTION DESIGN PARAMETERS:"
113  A1 = 1 + .365 * B1:D1 = .707 * B1:F1 = FZ * A1:F2 = FZ / A1
114  Q1 = 1 / D1
115  F1 =  INT (F1 * 10 + .5) / 10
116  PRINT : PRINT " CENTER FREQUENCY = ";F1;" HZ"; TAB( 33)"Q = "; INT (
     Q1 * 10 + .5) / 10
117  PRINT : PRINT "2ND SECTION DESIGN PARAMETERS:"
118  F2 =  INT (F2 * 10 + .5) / 10
119  PRINT : PRINT " CENTER FREQUENCY = ";F2;" HZ"; TAB( 33)"Q = "; INT (
     Q1 * 10 + .5) / 10
120  A$ = "Z":FI = FZ: GOTO 124
121  PRINT : INPUT "AMPLITUDE RESPONSE (Y/N) ";A$
122  IF A$ = "N" THEN 133
123  PRINT : INPUT "INPUT FREQUENCY IN HZ? ";FI
124  FA = FI / F1:FC = FI / F2:Q2 = Q1
125  DA =  SQR (1 + (Q1 ^ 2) * (((FA ^ 2) - 1) / FA) ^ 2)
126  DC =  SQR (1 + (Q2 ^ 2) * (((FC ^ 2) - 1) / FC) ^ 2)
127  Y = 1 / (DA * DC):DB = 8.68589 *  LOG (Y)
128  IF A$ <  > "Z" THEN 130
129  PRINT : PRINT "GAIN OF CASCADED STAGES MUST BE AT": PRINT "LEAST "; -
     INT (DB * 10 + .5) / 10;" DB": GOTO 121
130  PRINT : PRINT "   AMPLITUDE RESPONSE WITH INSERTION": PRINT "   LOSS
     = "; INT (DB * 10 + .5) / 10;" DB"
131  PRINT : INPUT "ANOTHER INPUT FREQUENCY (Y/N)? ";C$
132  IF C$ = "Y" THEN 123
133  PRINT : PRINT "DESIGN COMPLETED": GOTO 189
134  IF P = 4 THEN 158
135  PRINT : PRINT "1ST SECTION DESIGN PARAMETERS:"
136  A1 = 1 + .45 * B1:A3 = A1:D1 = .5 * B1:D3 = D1:D2 = 1 / B1:F2 = FZ
137  F1 = FZ * A1:F3 = FZ / A3:Q1 = 1 / D1:Q3 = 1 / D3:Q2 = D2
138  F1 =  INT (F1 * 10 + .5) / 10:F2 =  INT (F2 * 10 + .5) / 10:F3 =  INT
     (F3 * 10 + .5) / 10
139  PRINT : PRINT " CENTER FREQUENCY = ";F1;" HZ"; TAB( 33)"Q = "; INT (
     Q1 * 10 + .5) / 10
140  PRINT : PRINT "2ND SECTION DESIGN PARAMETERS:"
141  PRINT : PRINT " CENTER FREQUENCY = ";F2;" HZ"; TAB( 33)"Q = "; INT (
     Q2 * 10 + .5) / 10
142  PRINT : PRINT "3RD SECTION DESIGN PARAMETERS:"
143  PRINT : PRINT " CENTER FREQUENCY = ";F3;" HZ"; TAB( 33)"Q = "; INT (
     Q3 * 10 + .5) / 10
144  A$ = "Z":FI = F2: GOTO 148
```

Continued on next page.

Fig. 6-29. Listing for TUNEDBP program.

```
145  PRINT : INPUT "AMPLITUDE RESPONSE (Y/N)? ";A$
146  IF A$ = "N" THEN 157
147  PRINT : INPUT "INPUT FREQUENCY IN HZ? ";FI
148  FA = FI / F1:FB = FI / F2:FC = FI / F3:DA =  SQR (1 + (Q1 ^ 2) * (((F
     A ^ 2) - 1) / FA) ^ 2)
149  DC =  SQR (1 + (Q2 ^ 2) * (((FB ^ 2) - 1) / FB) ^ 2)
150  DD =  SQR (1 + (Q3 ^ 2) * (((FC ^ 2) - 1) / FC) ^ 2)
151  Y = 1 / (DA * DC * DD):DB = 8.68589 *  LOG (Y)
152  IF A$ <  > "Z" THEN 154
153  PRINT : PRINT "GAIN OF CASCADED STAGES MUST BE AT": PRINT "LEAST "; -
     INT (DB * 10 + .5) / 10;" DB": GOTO 145
154  PRINT : PRINT "   AMPLITUDE RESPONSE WITH INSERTION": PRINT "   LOSS
     = "; INT (DB * 10 + .5) / 10;" DB"
155  PRINT : INPUT "ANOTHER INPUT FREQUENCY (Y/N)? ";C$
156  IF C$ = "Y" THEN 147
157  PRINT : PRINT "DESIGN COMPLETED": GOTO 189
158  IF P <  > 4 THEN 187
159  PRINT : PRINT "1ST SECTION DESIGN PARAMETERS:"
160  A1 = 1 + .485 * B1:A3 = 1 + .195 * B1:D1 = .38 * B1:D3 = .923 * B1
161  F1 = FZ * A1:F2 = FZ * A3:F3 = FZ / A3:F4 = FZ / A1
162  Q1 = 1 / D1:Q2 = 1 / D3:Q3 = Q2:Q4 = Q1
163  F1 =  INT (F1 * 10 + .5) / 10:F2 =  INT (F2 * 10 + .5) / 10:F3 =  INT
     (F3 * 10 + .5) / 10:F4 =  INT (F4 * 10 + .5) / 10
164  PRINT : PRINT " CENTER FREQUENCY = ";F1;" HZ"; TAB( 33)"Q = "; INT (
     Q1 * 10 + .5) / 10
165  PRINT : PRINT "2ND SECTION DESIGN PARAMETERS:"
166  PRINT : PRINT " CENTER FREQUENCY = ";F2;" HZ"; TAB( 33)"Q = "; INT (
     Q2 * 10 + .5) / 10
167  PRINT : PRINT "3RD SECTION DESIGN PARAMETERS:"
168  PRINT : PRINT " CENTER FREQUENCY = ";F3;" HZ"; TAB( 33)"Q = "; INT (
     Q3 * 10 + .5) / 10
169  PRINT : PRINT "4TH SECTION DESIGN PARAMETERS:"
170  PRINT : PRINT " CENTER FREQUENCY = ";F4;" HZ"; TAB( 33)"Q = "; INT (
     Q4 * 10 + .5) / 10
171  A$ = "Z":FI = FZ: GOTO 175
172  PRINT : INPUT "AMPLITUDE RESPONSE (Y/N)? ";A$
173  IF A$ = "N" THEN 186
174  PRINT : INPUT "INPUT FREQUENCY IN HZ? ";FI
175  FA = FI / F1:FB = FI / F2:FC = FI / F3:FD = FI / F4
176  DA =  SQR (1 + (Q1 ^ 2) * (((FA ^ 2) - 1) / FA) ^ 2)
177  DC =  SQR (1 + (Q2 ^ 2) * (((FB ^ 2) - 1) / FB) ^ 2)
178  DD =  SQR (1 + (Q3 ^ 3) * (((FC ^ 2) - 1) / FC) ^ 2)
179  DE =  SQR (1 + (Q4 ^ 2) * (((FD ^ 2) - 1) / FD) ^ 2)
180  Y = 1 / (DA * DC * DD * DE):DB = 8.68589 *  LOG (Y)
181  IF A$ <  > "Z" THEN 183
182  PRINT : PRINT "GAIN OF CASCADED STAGES MUST BE AT": PRINT "LEAST "; -
     INT (DB * 10 + .5) / 10;" DB": GOTO 172
183  PRINT : PRINT "   AMPLITUDE RESPONSE WITH INSERTION": PRINT "   LOSS
     = "; INT (DB * 10 + .5) / 10;" DB"
184  PRINT : INPUT "ANOTHER INPUT FREQUENCY (Y/N)? ";C$
185  IF C$ = "Y" THEN 174
186  PRINT : PRINT "DESIGN COMPLETED": GOTO 189
187  INVERSE : PRINT : PRINT "DESIGNS LIMITED UP TO 4 STAGES": PRINT  TAB(
     10)"TRY AGAIN            ": NORMAL
188  GOTO 105
189  END
```

Fig. 6-29 (cont). Listing for TUNEDBP program.


```
100  REM  NOTCH FILTER DESIGN (NOTCH)
105  HOME : VTAB (7): PRINT  TAB( 10): INVERSE : PRINT "NOTCH FILTER DESI
     GN": NORMAL
110  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
115  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
120  FOR II = 1 TO 2500: NEXT II: HOME
125  PRINT
130  PRINT "DESIGN BASED ON KNOWING:": PRINT : PRINT "     1. Q"
135  PRINT : PRINT "     2. 3-DB BANDWIDTH"
140  PRINT : PRINT "     3. UPPER & LOWER 3-DB FREQUENCIES"
145  PRINT : INPUT "SELECT # ?";Z9$
150  IF Z9$ <  > "1" AND Z9$ <  > "2" AND Z9$ <  > "3" THEN 145
155  IF Z9$ = "1" THEN 170
160  IF Z9$ = "2" THEN 190
165  IF Z9$ = "3" THEN 200
170  PRINT : INPUT " Q ? ";Q
175  IF Q < 25 THEN 185
180  PRINT : INVERSE : PRINT "Q MUST BE LESS THAN 25.": NORMAL : GOTO 170
185  GOTO 235
190  PRINT : INPUT "3-DB BANDWIDTH IN HZ ";BW
195  GOTO 235
200  PRINT : INPUT "LOWER 3-DB FREQUENCY IN HZ ";FL
205  PRINT : INPUT "UPPER 3-DB FREQUENCY IN HZ ";FH
210  FO =  SQR (FH * FL)
215  PRINT : PRINT "   NOTCH (NULL) FREQUENCY = "; INT (FO * 10 + .5) / 1
     0;" HZ"
220  BW = FH - FL:Q = FO / BW
225  PRINT : PRINT "   3-DB BANDWIDTH = "; INT (BW * 10 + .5) / 10;" HZ"
230  PRINT : PRINT "   Q = "; INT (Q * 10 + .5) / 10
235  PRINT : PRINT "HIT RETURN TO CONTINUE";
240  CALL  - 756: CALL  - 936
245  HOME : PRINT : INPUT "STANDARD CAPACITOR VALUE IN UF? ";C
250  IF Z9$ = "1" THEN 260
255  IF Z9$ = "3" THEN 275
260  PRINT : INPUT "NOTCH (NULL) FREQUENCY IN HZ? ";FO
265  IF Z9$ = "1" THEN 275
270  Q = FO / BW
275  PRINT : INPUT "PASSBAND VOLTAGE GAIN? ";G
280  H = 100 / (FO * C):R1 = 0.796 * Q * H:R2 = R1 / ((Q ^ 2) - 1)
285  R3 = H:R4 = 4 * R1:R5 = 2 * H:R6 = G * H
290  R1 =  INT (R1 * 100 + .5) / 100:R2 =  INT (R2 * 100 + .5) / 100
295  R3 =  INT (R3 * 100 + .5) / 100
300  R4 =  INT (R4 * 100 + .5) / 100:R5 =  INT (R5 * 100 + .5) / 100
305  R6 =  INT (R6 * 100 + .5) / 100
310  PRINT : PRINT "THEN.....": PRINT : PRINT "R1= ";R1;" K-OHMS"
315  PRINT : PRINT "R2= ";R2;" K-OHMS": PRINT : PRINT "R3= ";R3;" K-OHMS"
320  PRINT : PRINT "R4= ";R4;" K-OHMS": PRINT : PRINT "R5= ";R5;" K-OHMS"
325  PRINT : PRINT "R6= ";R6;" K-OHMS"
330  PRINT : INPUT "TRY ANOTHER SET OF VALUES (Y/N) ";X$
335  IF X$ = "Y" THEN 245
340  PRINT : PRINT "ENTER STANDARD COMPONENT VALUES FOR:"
345  PRINT : INPUT "R1 (IN K-OHMS)? ";R1
350  INPUT "R2 (IN K-OHMS)? ";R2
355  INPUT "R3 (IN K-OHMS)? ";R3
360  INPUT "R4 (IN K-OHMS)? ";R4
365  INPUT "R6 (IN K-OHMS)? ";R6
```

Continued on next page.

Fig. 6-33. Listing for NOTCH program.

```
370 INPUT "C (IN UF)? ";C
375 PRINT : PRINT "BASED ON THE ABOVE STANDARD VALUES:"
380 R2 = 1E3 * R2:R1 = 1E3 * R1:C = 1E - 6 * C:R4 = 1E3 * R4
385 FO =  SQR (((R2 + R1) / (R1 * R2)) * (1 / (R4 * (C ^ 2)))) / 6.28
390 H = 1E - 4 / (FO * C):Q = R1 * 1E - 3 / (.796 * H):G = R6 / R3
395 PRINT : PRINT "NOTCH FREQUENCY = "; INT (FO * 100 + .5) / 100;" HZ"
400 PRINT : PRINT "PASSBAND VOLTAGE GAIN = "; INT (G * 100 + .5) / 100
405 PRINT : PRINT "Q = "; INT (Q * 100 + .5) / 100: PRINT : PRINT "BANDW
    IDTH = "; INT ((FO / Q) * 100 + .5) / 100;" HZ"
410 PRINT : INPUT "ANOTHER SET OF STANDARD VALUES (Y/N)? ";X$
415 IF X$ = "Y" THEN 340
420 HOME : PRINT : PRINT "DO YOU WISH TO KNOW THE AMPLITUDE": INPUT "RES
    PONSE (Y/N)? ";X$
425 IF X$ = "N" THEN 465
430 PRINT : INPUT "ENTER INPUT FREQUENCY IN HZ? ";FI
435 F = FI / FO:V = 4.34294 *  LOG ((1 - F ^ 2) ^ 2) - 4.34294 *  LOG ((1
    - F ^ 2) ^ 2 + F / Q ^ 2)
440 DB = (20 *  LOG (G) /  LOG (10)) + V
445 DB =  INT (DB * 100 + .5) / 100
450 PRINT : PRINT "   AMPLITUDE RESPONSE = ";DB;" DB"
455 PRINT : INPUT "TRY ANOTHER FREQUENCY (Y/N)? ";X$
460 IF X$ = "Y" THEN 430
465 PRINT : PRINT "DESIGN COMPLETED"
470 END
```

Fig. 6-33 (cont). Listing for NOTCH program.

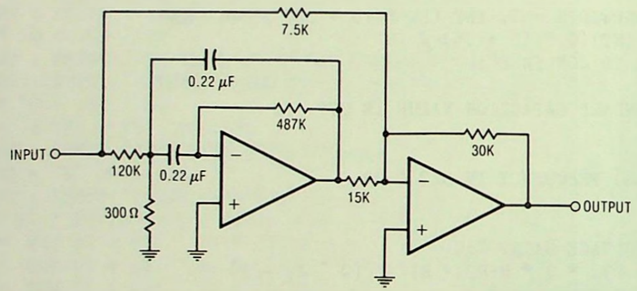


Fig. 6-35. Final circuit for Example 6-7.

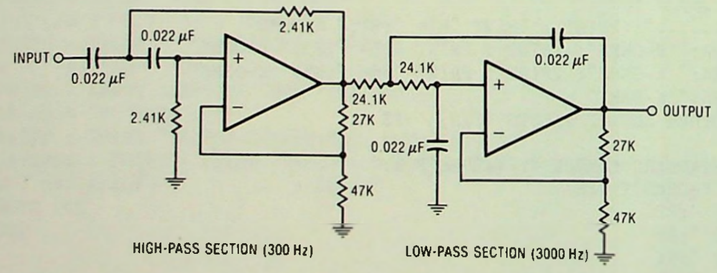


Fig. 6-36. Final circuit for Example 6-8.


```
                NOTCH FILTER DESIGN

DESIGN BASED ON KNOWING:
     1. Q
     2. 3-DB BANDWIDTH
     3. UPPER & LOWER 3-DB FREQUENCIES

SELECT # ?1

Q ? 20

HIT RETURN TO CONTINUE
STANDARD CAPACITOR VALUE IN UF? .22
NOTCH (NULL) FREQUENCY IN HZ? 60
PASSBAND VOLTAGE GAIN? 4


THEN.......
R1= 120.61 K-OHMS
R2= .3 K-OHMS
R3= 7.58 K-OHMS
R4= 482.42 K-OHMS
R5= 15.15 K-OHMS
R6= 30.3 K-OHMS

TRY ANOTHER SET OF VALUES (Y/N) N
ENTER STANDARD COMPONENT VALUES FOR:

 R1 (IN K-OHMS)? 120
 R2 (IN K-OHMS)? .300
 R3 (IN K-OHMS)? 7.5
 R4 (IN K-OHMS)? 487
 R6 (IN K-OHMS)? 30
 C (IN UF)? .22

BASED ON THE ABOVE STANDARD VALUES:
NOTCH FREQUENCY = 59.96 HZ
PASSBAND VOLTAGE GAIN = 4
Q = 19.89
BANDWIDTH = 3.02 HZ

ANOTHER SET OF STANDARD VALUES (Y/N)? N

DO YOU WISH TO KNOW THE AMPLITUDE
RESPONSE (Y/N)? Y

ENTER INPUT FREQUENCY IN HZ? 59.96
     AMPLITUDE RESPONSE = -40.14 DB

TRY ANOTHER FREQUENCY (Y/N)? N

DESIGN COMPLETED
```

Fig. 6-34. Results for Example 6-7.

for the first and third sections is 4, while the gain for the second section is 2, giving an overall gain of 32 (+30.1 dB).

THE NOTCH FILTER

Fig. 6-32 shows a notch filter for values of Q less than 25. For values of Q greater than 25, the state-variable notch filter should be used. As with all the filter designs discussed, both capacitors are made equal to each other.

The NOTCH program listed in Fig. 6-33 requires the following initial information:

1. Design determined by knowing either Q, 3-dB bandwidth, or the 3-dB frequencies.
2. Standard capacitor value.
3. Notch, or null frequency.
4. Passband voltage gain.

after which the remaining resistor values are determined.

Example 6-7

Design a 60-Hz notch filter with a Q of 20 and a passband gain of 4 (+12.04 dB).

Selecting a standard capacitor value of 0.22 μF, Fig. 6-34 shows the program's output, while Fig. 6-35 illustrates the final circuit.

A WIDEBAND FILTER

Very often, particularly in audio applications, it is desired to pass a wide band of frequencies such that the filter's bandwidth is to be more than twice its center frequency. In addition, the passband gain is to be relatively constant. Such is the case of a filter that passes audio frequencies ranging from 300 Hz to 3000 Hz.

The "center frequency" is then 949 Hz, while the bandwidth is 2700 Hz. Virtually no standard bandpass filter (including stagger-tuned designs) will function properly over this range. However, by cascading a 3000-Hz low-pass section with a 300-Hz high-pass section (from second- to sixth-order), we are then able to produce the desired response, as shown in the following example.

Example 6-8

Design a 300- to 3000-Hz wideband filter having relatively constant passband gain with second-order stopband responses.

Since the passband gain is to be essentially constant, then a second-order Butterworth design should be used for each section. Using the LPHP program discussed earlier in this chapter, we can then design each section individually, as shown by the final circuit of Fig. 6-36.

7

Solid-State Devices

This chapter describes programs which simplify the design of zener diode voltage regulators, the single stage common-emitter transistor amplifier, the analysis of transistor biasing, heat sinks, and 555 timer circuits.

ZENER DIODE VOLTAGE REGULATOR DESIGN

The ZENER program listing of Fig. 7-1 allows for the determination of the proper resistance and zener diode power rating for the regulator circuit of Fig. 7-2. When executed, the program requires the following specifications:

1. Minimum and maximum dc input voltage.
2. Minimum and maximum load current, in milliamperes.
3. Zener diode (regulated output) voltage.

In addition to determining the series resistance and the minimum power rating for the zener diode, the program also determines the diode rating if the load is disconnected from the regulator, so that the load current is zero.

```
100  REM   ZENER VOLTAGE REGULATOR DESIGN (ZENER)
105  HOME : CLEAR : VTAB (7): PRINT  TAB( 2): INVERSE : PRINT "ZENER DIOD
     E VOLTAGE REGULATOR DESIGN": NORMAL
110  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
115  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
120  FOR I = 1 TO 2500: NEXT
125  HOME : INPUT "MINIMUM DC INPUT VOLTAGE? ";VN
130  PRINT : INPUT "MAXIMUM DC INPUT VOLTAGE? ";VX
135  PRINT : INPUT "MINIMUM LOAD CURRENT (IN MA)? ";IN
140  PRINT : INPUT "MAXIMUM LOAD CURRENT (IN MA)? ";IX
145  IN = IN * 1E - 3:IX = IX * 1E - 3
150  PRINT : INPUT "ZENER DIODE VOLTAGE? ";VZ
155  R = (VN - VZ) / (1.1 * IX)
160  P = VZ * (((VX - VZ) / R) - IN)
165  R =  INT (R * 10 + .5) / 10:P =  INT (P * 100 + .5) / 100
170  PRINT : PRINT : PRINT "THEN...."
175  PRINT : PRINT "REQUIRED SERIES RESISTANCE  ": PRINT  SPC( 9)R;" OHMS
     "
180  PRINT : PRINT "MINIMUM ZENER DIODE RATING ": PRINT  SPC( 9)P;" WATTS
     "
185  IF IN = 0 THEN 200
190  Z = VZ * ((VX - VZ) / R):Z =  INT (Z * 100 + .5) / 100
195  PRINT : PRINT "MINIMUM ZENER DIODE RATING  (NO LOAD)": PRINT  SPC( 9
     )Z;" WATTS"
200  PRINT : END
```

Fig. 7-1. Listing for ZENER program.

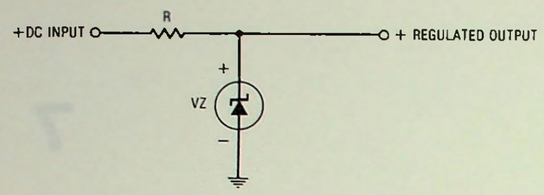


Fig. 7-2. Zener diode voltage regulator.

Example 7-1

Determine the necessary series resistance and diode rating for a regulator circuit having a dc input voltage from 11.95 to 12.05 volts, and connected to a load which draws between 75 and 300 mA.

The output results are shown in Fig. 7-3.

555 TIMER DESIGN

The 555 integrated circuit timer is one of the most popular devices in use. With the same device, it is possible to design both monostable (one-shot) and astable (free-running) multivibrator circuits, as shown in Figs. 7-4 and 7-5 respectively.

The TIMER555 program listing of Fig. 7-6 determines the required component values for monostable and astable circuits, as well as displaying the final schematic circuit on the video display. Depending on the type of design, the following initial information is required:

1. Monostable Design:
   a. Standard capacitor value.
   b. Time delay in seconds.
2. Astable Design:
   a. Standard capacitor value.
   b. Output frequency in Hz.
   c. Percent duty cycle.

```
      ZENER DIODE VOLTAGE REGULATOR DESIGN
MINIMUN DC INPUT VOLTAGE? 11.95
MAXIMUM DC INPUT VOLTAGE? 12.05
MINIMUM LOAD CURRENT (IN MA)? 75
MAXIMUM LOAD CURRENT (IN MA)? 300
ZENER DIODE VOLTAGE? 5.1

THEN....

REQUIRED SERIES RESISTANCE
         20.8 OHMS

MINIMUM ZENER DIODE RATING
         1.33 WATTS

MINIMUM ZENER DIODE RATING  (NO LOAD)
         1.7 WATTS
```

Fig. 7-3. Results for Example 7-1.

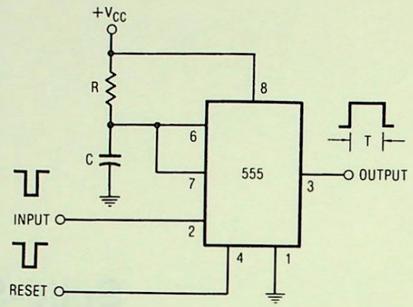


Fig. 7-4. Monostable multivibrator using the 555 timer.

The percent duty cycle must be greater than 50% and less than 100%. If the duty cycle is to be symmetrical, then use the value of 50.1% for the duty cycle. For both designs, the program permits the calculated component values to be changed if they are not close enough to standard values.

Example 7-2

Using a 555 timer, design a monostable multivibrator having a time delay of 20 seconds.

As shown in Fig. 7-7A, we started by choosing a value of 50 μF. However the computed resistance value did not appear to be close to a standard value. The process was repeated until the final value of 150 μF was used so that a 120 kilohm resistor could be used. The final circuit is shown in Fig. 7-7B.

Example 7-3

Using a 555 timer, design an astable multivibrator with an output frequency of 400 Hz and a 60% duty cycle. The output results are shown in Fig. 7-8.

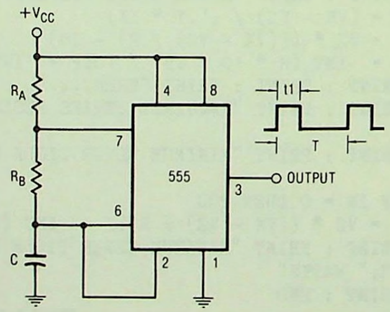


Fig. 7-5. Astable multivibrator using the 555 timer.


```
100  REM   ** 555 TIMER DESIGN **
105  CALL  - 936: VTAB (7): PRINT  TAB( 2): INVERSE : PRINT "555 INTEGRAT
     ED CIRCUIT TIMER DESIGN": NORMAL
110  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980"
115  PRINT : PRINT  TAB( 9)"HOWARD M. BERLIN, LTD."
120  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
125  FOR I = 1 TO 2000: NEXT I
130  HOME : VTAB 10
135  PRINT "<M>ONOSTABLE OR <A>STABLE MODE ";: INPUT C$
140  IF C$ <  > "M" AND C$ <  > "A" THEN 105
145  IF C$ = "A" GOTO 200
150  CALL  - 936: PRINT : PRINT  TAB( 4)"555 TIMER - MONOSTABLE OPERATION
     "
155  PRINT : INPUT "TIME DELAY (IN SECONDS)? ";T
160  IF T > 1E - 3 THEN 170
165  INVERSE : PRINT : PRINT "DELAY MUST BE > .001 SECONDS": NORMAL : GOTO
     155
170  PRINT : INPUT "C (UF)? ";C
175  IF C < .001 THEN 170
180  R =  INT (((1E3 * T) / (1.1 * C)) * 100 + .5) / 100: PRINT : PRINT "
     R = ";R;" K-OHMS"
185  PRINT : INPUT "CHANGE VALUES (Y/N) ";A$: IF A$ = "Y" THEN 170
190  GOSUB 280
195  END
200  CALL  - 936: PRINT : PRINT  TAB( 10)"555 TIMER - ASTABLE OPERATION"
205  PRINT : INPUT "OUTPUT FREQUENCY (HZ)? ";F
210  IF F > 200E3 THEN 1 0
215  PRINT : INPUT "PERCENT DUTY CYCLE (>50%)? ";D:D = D / 100
220  IF D <  = .5 THEN 215
225  PRINT : INPUT "C (UF)? ";C:C = C * 1E - 6
230  IF C < 1E - 9 THEN 225
235  RA = (2 * D - 1) / ( LOG (2) * C * F)
240  RB = (1 / (2 *  LOG (2) * C * F)) - (RA / 2)
245  X = C * (RA + 2 * RB):F = (1 /  LOG (2)) * 1 / X
250  RA =  INT ((RA / 1E3) * 100 + .5) / 100:RB =  INT ((RB / 1E3) * 100 +
     .5) / 100:C = C / 1E - 6
255  D = D * 100: PRINT : PRINT "RA = ";RA;" K-OHMS";"  RB = ";RB;" K-OHMS
     "
260  PRINT : INPUT "CHANGE INPUT VALUES (Y/N)? ";B$
265  IF B$ <  > "N" THEN 215
270  GOSUB 380
275  VTAB (23): END
280  REM  SUBROUTINE TO DRAW TIMER
285  CALL  - 936: FOR I = 1 TO 44: READ P,Z
290  IF P = 0 THEN 320
295  POKE P,Z: NEXT I
300  DATA  1037,171,1038,214,1039,195,1040,195,1047,171,1048,214,1049,195
     ,1050,195
305  DATA  1153,6,1154,9,1155,14,1156,1,1157,12,1281,4,1282,5,1283,19,128
     4,9,1285,7,1286,14,1428,180,1434,184
310  DATA  1804,210,1811,182,1348,179,1598,53,1599,53,1600,53,1714,195,17
     15,171,1722,183
315  DATA  1736,207,1737,213,1738,212,1739,208,1740,213,1741,212,1636,178
     ,1883,201,1884,206,1885,208,1886,213,1887,212,1897,177,0,0
320  FOR I = 1 TO 1000: READ P
325  IF P = 0 THEN 365
330  POKE P,32: NEXT I
```

Continued on next page.

Fig. 7-6. Listing for TIMER555.

```
335 DATA  1166,1176,1294,1304,1421,1422,1423,1430,1431,1432,1549,1551,15
    58,1560,1677,1679,1686,1688,1805,1806,1807,1812,1813,1814,1815,1816,
    1817,1818
340 DATA  1934,1940,1946,1078,1079,1080,1081,1082,1083,1084,1090,1206,12
    12,1218,1334,1340,1346,1462,1463,1464,1465,1466,1467,1468,1474
345 DATA  1590,1596,1602,1603,1604,1605,1606,1718,1724,1730,1844,1845,18
    46,1847,1848,1852,1858,1980,1986,1116,1117,1118,1119,1120,1124,1130,
    1246,1252,1258
350 DATA  1374,1380,1381,1382,1383,1384,1385,1386,1500,1501,1502,1503,15
    04,1510,1512,1628,1630,1632,1638,1640,1641,1642,1643,1644,1645,1646,
    1647
355 DATA  1766,1775,1889,1890,1891,1892,1893,1894,1901,1902,1903,1904,19
    05,2029,2031,2033,1749,1750,1752,1753,1878,1880,2006,2007,2008
360 DATA  1225,1226,1227,1228,1353,1356,1480,1481,1484,1485,0,0
365 VTAB (9): PRINT R;"K-OHMS": VTAB (16): PRINT  TAB( 4)C;"UF";: PRINT
    TAB( 30)T;" SEC": POKE 1980,32: POKE 1986,32
370 VTAB (23)
375 RETURN
380 FOR I * 1 TO 234: READ P: NEXT I
385 CALL  - 936: VTAB (7): PRINT RA;"K-OHMS"
390 VTAB (14): PRINT RB;"K-OHMS";: PRINT  TAB( 33)F;" HZ"
395 VTAB (18): PRINT  TAB( 31)"*   ";D;"%": VTAB (19): PRINT  TAB( 1)C;"
    UF"
400 FOR I * 1 TO 1000: READ P,Z
405 IF P *  - 1 THEN 435
410 POKE P,Z: NEXT I
415 DATA  1036,171,1037,214,1038,195,1039,195,1047,171,1048,214,1049,195
    ,1050,195,1153,6,1154,9,1155,14,1156,1,1157,12
420 DATA  1281,4,1282,5,1283,19,1284,9,1285,7,1286,14,1428,180,1434,184,
    1545,210,1546,193,1210,183,1348,179,1457,210,1458,194
425 DATA  1480,207,1481,213,1482,212,1483,208,1484,213,1485,212,1594,182
    ,1598,53,1599,53,1600,53,1988,196,1989,213,1990,212
430 DATA  1991,217,1993,195,1994,217,1995,195,1996,204,1997,197,1113,195
    ,1114,171,1636,178,1897,177,-1,-1
435 FOR J * 1 TO 1000: READ H
440 IF H *  - 2 THEN 485
445 POKE H,32: NEXT J
450 DATA  1165,1176,1292,1293,1294,1304,1420,1422,1430,1431,1432,1548,15
    50,1558,1560,1676,1677,1678,1686,1688,1805,1812,1813,1814,1815,1816,
    1817,1818
455 DATA  1933,1934,1935,1936,1937,1938,1939,1940,1946,1077,1084,1090,12
    04,1205,1206,1212,1218,1332,1334,1340,1346,1460,1462,1468,1474
460 DATA  1588,1589,1590,1596,1602,1603,1604,1605,1606,1717,1724,1730,18
    45,1846,1847,1848,1849,1850,1851,1852,1858,1973,1117,1122,1124,1130
465 DATA  1243,1244,1245,1246,1247,1250,1252,1258,1378,1380,1381,1382,13
    83,1384,1385,1386,1499,1500,1501,1502,1503,1506,1510,1512
470 DATA  1629,1634,1638,1640,1641,1642,1643,1644,1645,1646,1647,1757,17
    62,1766,1775,1883,1884,1885,1886,1887,1890,1891,1892,1893,1894
475 DATA  1901,1902,1903,1904,1905,2011,2013,2015,2029,2031,2033
480 DATA  1978,1980,1986,-2
485 RETURN
```

Fig. 7-6 (cont). Listing for TIMER555.

TRANSISTORS

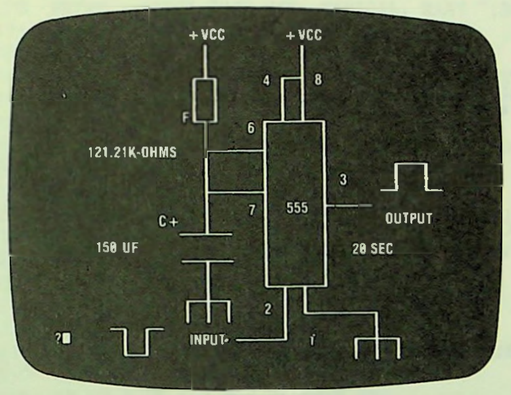
In these days of integrated circuits that virtually do everything, the transistor is still an important facet of the design and analysis of electronic circuits. In general, the one-power supply bias network of a bipolar transistor will have the generalized circuit of Fig. 7-9, in that one or more resistors will be either open or short circuited.

For any given circuit, the BIAS program of Fig. 7-10 will determine the operating point of the transistor with the values of the three terminal currents, and the quiescent output voltage, V_{CO}. When run, the required parameters are:

1. Supply voltage.
2. Typical beta of transistor.


```
   555 TIMER - MONOSTABLE OPERATION
TIME DELAY (IN SECONDS)? 20

C (UF)? 50
     R = 363.64 K-OHMS

CHANGE VALUES (Y/N) Y

C (UF)? 100
     R = 181.82 K-OHMS

CHANGE VALUES (Y/N) Y

C (UF)? 150
     R = 121.21 K-OHMS

CHANGE VALUES (Y/N) N
```

(A) Output results.



(B) Video plot.

Fig. 7-7. Results for Example 7-2.

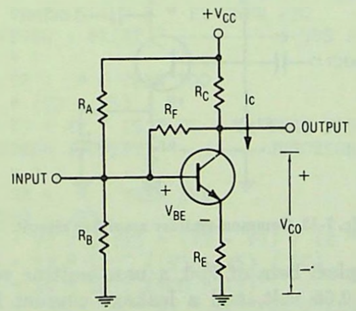
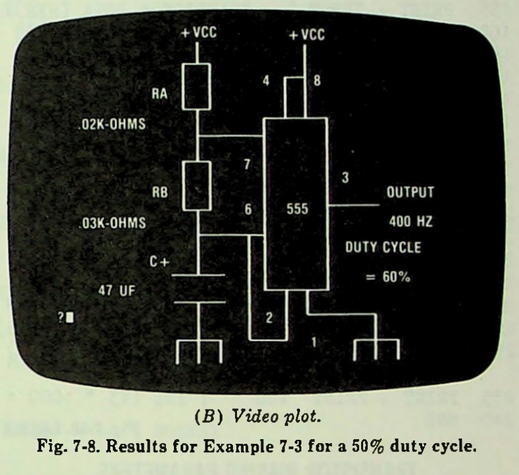


Fig. 7-9. Generalized transistor bias circuit.

```
   555 TIMER - ASTABLE OPERATION
OUTPUT FREQUENCY (HZ)? 400
PERCENT DUTY CYCLE (.50%)? 55
C (UF)? 33
RA = .01 K-OHMS  RB = .05 K-OHMS

CHANGE INPUT VALUES (Y/N)? Y

PERCENT DUTY CYCLE (.50%)? 60
C (UF)? 47
RA = .02 K-OHMS  RB = .03 K-OHMS

CHANGE INPUT VALUES (Y/N)? N
```

(A) Output results.



(B) Video plot.

Fig. 7-8. Results for Example 7-3 for a 50% duty cycle.

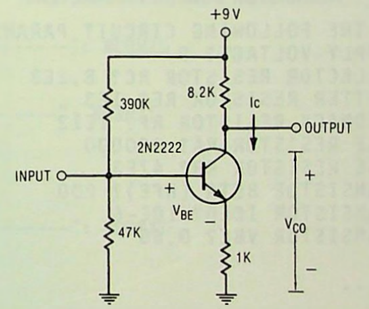


Fig. 7-11. Circuit for Example 7-4.

3. Leakage current I_{ceo}.
4. The base-emitter voltage, V_{be}.
5. The network resistors (in ohms).

If any one of the five resistors is represented by an open circuit, as is frequently the case with the feedback resistor RF, then the value of 1E12 (10^{12}) should be used.

```
100  REM   TRANSISTOR BIASING (BIAS)
105  CLEAR : HOME : VTAB (7): PRINT  TAB( 5): INVERSE : PRINT "TRANSISTOR
     BIASING PARAMETERS": NORMAL
110  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
115  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
120  FOR I = 1 TO 2500: NEXT : HOME
125  HOME : PRINT "ENTER THE FOLLOWING CIRCUIT PARAMETERS:": PRINT : INPUT
     "   SUPPLY VOLTAGE? ";V
130  PRINT : INPUT "   COLLECTOR RESISTOR RC? ";RC
135  PRINT : INPUT "   EMITTER RESISTOR RE? ";RE
140  PRINT : INPUT "   FEEDBACK RESISTOR RF? ";RF
145  PRINT : INPUT "   BASE RESISTOR RA? ";RA
150  PRINT : INPUT "   BASE RESISTOR RB? ";RB
155  PRINT : INPUT "   TRANSISTOR BETA (HFE)? ";B
160  PRINT : INPUT "   TRANSISTOR ICEO? ";IO
165  PRINT : INPUT "   TRANSISTOR VBE? ";V2
170  R1 = RA * RB / (RA + RB)
175  V1 = RB * V / (RA + RB)
180  R = RC + RF
185  RR = RE * R * IO / B + R1 * (V - V2)
190  Q = R1 * RE * IO / B + R * (V1 - V2 + IO * R1 / B)
195  S = R1 * RE * (1 + 1 / B) + R1 * R / B + RC * R1
200  T = RE * R * (1 + 1 / B)
205  U = (Q + RR) / (S + T)
210  V3 = V - RC * U
215  HOME : PRINT : PRINT " THEN..."
220  PRINT : PRINT "IC = "; INT ((U * 1000) * 1000 + .5) / 1000;" MA"
225  PRINT : PRINT "IB = "; INT ((U / B * 1E6) * 1000 + .5) / 1000;" UA"
230  PRINT : PRINT "IE = "; INT ((B / (B - 1) * U * 1000) * 1000 + .5) /
     1000;" MA"
235  PRINT : PRINT "VCO = "; INT (V3 * 1000 + .5) / 1000;" VOLTS"
240  END
```

Fig. 7-10. Listing for BIAS program.

```
        TRANSISTOR BIASING PARAMETERS

ENTER THE FOLLOWING CIRCUIT PARAMETERS:
   SUPPLY VOLTAGE? 9
   COLLECTOR RESISTOR RC? 8.2E3
   EMITTER RESISTOR RE? 1E3
   FEEDBACK RESISTOR RF? 1E12
   BASE RESISTOR RA? 390000
   BASE RESISTOR RB? 47E3
   TRANSISTOR BETA (HFE)? 200
   TRANSISTOR ICEO? 10E-6
   TRANSISTOR VBE? 0.65

 THEN...

IC = .264 MA
IB = 1.318 UA
IE = .265 MA
VCO = 6.839 VOLTS
```

Fig. 7-12. Results for Example 7-4.

Example 7-4

For the circuit of Fig. 7-11, determine the operating point of the 2N2222 transistor, assuming a typical beta of 200, a base-emitter voltage drop of 0.65 volt, and a leakage current I_{ceo} of 10 microamperes. The computed results are shown in Fig. 7-12.

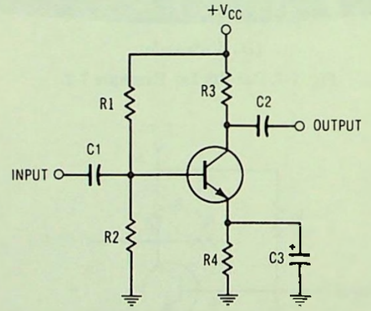


Fig. 7-13. Common-emitter amplifier circuit.

For designing a single stage common-emitter amplifier, as shown in Fig. 7-13, the TRANSAMP program of Fig. 7-14 is used. Having a particular transistor in mind, the program requires the following parameters:


```
100  REM  TRANSISTOR COMMON EMITTER AMPLIFIER DESIGN (TRANSAMP)
105  HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "TRANSISTOR COMMON
     EMITTER": NORMAL
110  VTAB (9): PRINT  TAB( 12): INVERSE : PRINT "AMPLIFIER DESIGN": NORMAL
115  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD"
120  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
125  FOR I = 1 TO 2500: NEXT I
130  HOME
135  INPUT "MAXIMUM TRANSISTOR VCE? ";V1: PRINT : INPUT "MAXIMUM COLLECTO
     R CURRENT (IN MA)? ";I1
140  PRINT : INPUT "MAXIMUM POWER DISSIPATION IN MW? ";P1
145  PRINT : INPUT "MAXIMUM JUNCTION TEMP IN DEG. C? ";T2
150  IF T2 > 0 THEN 160
155  PRINT : INPUT "TRANSISTOR THERMAL RESISTANCE?";T1
160  PRINT : INPUT "ICO IN UA? ";I2
165  I2 = I2 / 100
170  PRINT : INPUT "HFE MIN, TYPICAL, MAX? ";B1,B2,B3
175  PRINT : INPUT "TRANSISTOR VBE? ";V3
180  PRINT : INPUT "SUPPLY VOLTAGE? ";V2
185  PRINT : INPUT "QUIESCENT COLLECTOR CURRENT IN MA? ";I5
190  PRINT : INPUT "%VCC ACROSS EMITTER RESISTOR? ";V5
195  V5 = V5 * V2 / 100
200  PRINT : INPUT "AMBIENT TEMPERATURE IN DEG. C? ";T3
205  PRINT : INPUT "MAX % CHANGE FOR IC? ";I4
210  PRINT : INPUT "HIE IN K-OHMS? ";Z3
215  PRINT : INPUT "LOW FREQUENCY 3-DB POINT IN HZ? ";F1
220  HOME :HX = 0
225  IF V2 < 0.9 * V1 THEN 240
230  INVERSE : PRINT "---------SUPPLY VOLTAGE TOO HIGH---------": NORMAL
235  HX = 1
240  IF I5 < 0.4 * I1 THEN 255
245  INVERSE : PRINT "------COLLECTOR CURRENT TOO HIGH------": NORMAL
250  HX = 1
255  IF V2 * I5 < 0.8 * P1 THEN 275
260  PRINT
265  INVERSE : PRINT "--------POWER DISSIPATION TOO HIGH--------": NORMAL
270  HX = 1
275  IF V2 * I5 < 0.5 * P1 THEN 290
280  INVERSE : PRINT "------------USE A HEAT SINK------------": NORMAL
285  HX = 1
290  IF T2 <  = 0 THEN 300
295  T1 = (T2 - 25) / P1
300  IF T3 + V2 * I5 * T1 < T2 THEN 315
305  INVERSE : PRINT "---------JUNCTION TEMP TOO HIGH---------": NORMAL
310  HX = 1
315  IF HX <  > 1 THEN 325
320  PRINT : END
325  R4 = V5 / I5:R3 = (V2 - V5) / (2 * I5)
330  I3 = I2 * (2 ^ ((T3 - 25) / 10) - 1)
335  S1 = I4 * I5 / (100 * I3)
340  S3 = (B3 + 1) * B1 * S1 * I3 / (I5 * (B3 - B1))
345  S2 = S1 * S3 / (S1 + S3)
350  R5 = (1 + B2) * (S2 - 1) * R4 / (B2 + 1 - S2)
355  R1 = R5 * V2 / (V5 + V3)
360  R2 = V2 * R5 / (V2 - V5 - V3)
365  Z1 = R5 * Z3 / (R5 + Z3)
370  R4 = R4 * 1000
```

Continued on next page.

Fig. 7-14. Listing for TRANSAMP program.

```
375  PI = 3.1415927
380  C1 = 5E3 / (PI * F1 * Z1)
385  C2 = 5E6 / (PI * R4 * F1)
390  C3 = 5E3 / (PI * F1 * R3)
395  A1 = B2 * R3 / Z3
400  A2 = A1 * A1 + Z3 / R3
405  A3 = 10 *  LOG (A2) /  LOG (10)
410  A2 =  INT (A2 * 10 + .5) / 10
415  HOME : PRINT : PRINT "DESIGN SPECS:"
420  PRINT : PRINT "R1 = "; INT (R1 * 100 + .5) / 100;" K-OHMS": PRINT "R
     2 = "; INT (R2 * 100 + .5) / 100;" K-OHMS"
425  PRINT "R3 = "; INT (R3 * 100 + .5) / 100;" K-OHMS": PRINT "R4 = ";R4
     ;" OHMS"
430  PRINT "C1 = "; INT (C1 * 10 + .5) / 10;" UF"
435  PRINT "C2 = "; INT (C2 * 10 + .5) / 10;" UF"
440  PRINT "C3 = "; INT (C3 * 10 + .5) / 10;" UF"
445  V4 = .35 * (V2 - V5)
450  V6 = V4 / A1
455  PRINT "AV = ";A1
460  PRINT "AP = ";A2;"     "; INT (A3 * 10 + .5) / 10;" DB"
465  PRINT "ZIN = "; INT (Z1 * 100 + .5) / 100;" K-OHMS"
470  PRINT "ZOUT = "; INT (R3 * 100 + .5) / 100;" K-OHMS"
475  PRINT "VIN MAX = "; INT (V6 * 1000 + .5) / 1000
480  PRINT "VOUT MAX = "; INT (V4 * 1000 + .5) / 1000
485  POKE  - 16368,0
490  PRINT : INPUT "CHANGE PARAMETERS (Y/N)? ";ZZ$
495  IF ZZ$ = "Y" THEN 130
500  PRINT : PRINT "DESIGN COMPLETED": END
```

Fig. 7-14 (cont). Listing for TRANSAMP program.

1. Maximum collector-emitter voltage V_{CE}.
2. Maximum collector current, in milliamperes.
3. Maximum power dissipation, in milliwatts.
4. Maximum junction temperature, in degrees Celsius.
5. Leakage current, I_{co}, in microamperes.
7. Input impedance, h_{ie}.
8. Minimum, typical, and maximum beta (h_{FE}).

In addition, the following design constraints are required:

1. Maximum ambient temperature, in degrees Celsius.
2. Supply voltage.
3. Quiescent collector current, in milliamperes.
4. Percentage of the supply voltage across the emitter resistor.
5. Maximum percentage change for the collector current.
6. Low-frequency 3-dB point.

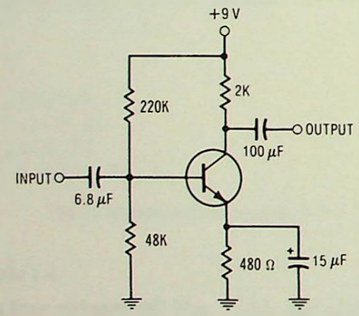


Fig. 7-16. Final circuit for Example 7-5.

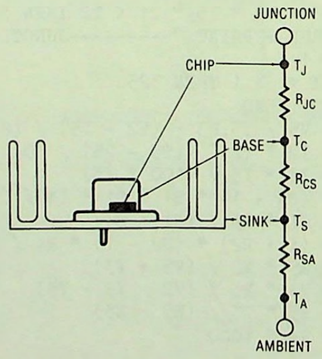


Fig. 7-17. Transistor heat sink, thermal model.


```
MAXIMUM TRANSISTOR VCE? 45
MAXIMUM COLLECTOR CURRENT (IN MA)? 30
MAXIMUM POWER DISSIPATION IN MW? 300
MAXIMUM JUNCTION TEMPERATURE IN DEG. C? 175
ICO IN UA? .002
HFE MIN, TYPICAL, MAX? 60,150,350
TRANSISTOR VBE? .65
SUPPLY VOLTAGE? 9
QUIESCENT COLLECTOR CURRENT IN MA? 2
%VCC ACROSS EMITTER RESISTOR? 10
AMBIENT TEMPERATURE IN DEG. C? 30
MAX % CHANGE FOR IC? 75
HIE IN K-OHMS? 5
LOW FREQUENCY 3-DB POINT IN HZ? 50


DESIGN SPECS:

R1 = 218.42 K-OHMS
R2 = 45.44 K-OHMS
R3 = 2.03 K-OHMS
R4 = 450 OHMS
C1 = 7.2 UF
C2 = 70.7 UF
C3 = 15.7 UF
AV = 60.75
AP = 3693      35.7 DB
ZIN = 4.41 K-OHMS
ZOUT = 2.03 K-OHMS
VIN MAX = .047
VOUT MAX = 2.835

CHANGE PARAMETERS (Y/N)? N

DESIGN COMPLETED
```

Fig. 7-15. Results for Example 7-5.

When executed, the program determines the value for the four resistors and three capacitors, the ac voltage and power gains, the input and output impedances, and the maximum input and output voltages of the amplifier.

Example 7-5

For the circuit of Fig. 7-13, design a common-emitter amplifier operating from a 9-volt supply with a minimum input frequency of 50 Hz. For a particular transistor, the parameters are:

Maximum V_{CE} = 45 volts.
Maximum collector current = 30 mA.
Maximum power dissipation = 300 mW.
Maximum junction temperature = 175 degrees Celsius.
I_{co} = .002 μA.
Minimum beta = 60.
Typical beta = 150.
Maximum beta = 350.
V_{be} = 0.65 volt.
h_{ie} = 5 kilohms.

In addition, the quiescent collector current is to be 2 mA, the collector current is to vary up to a maximum of 75%, and the maximum expected ambient temperature is 30 degrees Celsius.

The computed output results are shown in Fig. 7-15, giving the resultant circuit of Fig. 7-16.

HEAT SINK DESIGN

Selecting the proper heat sink is just as important as the design of transistor or integrated circuits. Using the thermal model shown in Fig. 7-17, the HEATSINK program of Fig. 7-18 determines the proper heat sink required when the following parameters are specified:

1. Power dissipation of the device.
2. Maximum junction temperature.

```
100  REM  HEATSINK DESIGN PROGRAM (HEATSINK)
105  CALL  - 936: VTAB (7): PRINT  TAB( 12): INVERSE : PRINT "HEAT SINK D
     ESIGN": NORMAL
110  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
115  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
120  FOR I = 1 TO 2500: NEXT I: CALL  - 936
125  PRINT : PRINT "ALL TEMPERATURES ARE TO BE EXPRESSED IN": PRINT : PRINT
     TAB( 11): FLASH : PRINT "DEGREES C": NORMAL
130  PRINT : INPUT "REQUIRED POWER DISSIPATION (WATTS)? ";P
135  PRINT : INPUT "MAXIMUM AMBIENT TEMPERATURE? ";TA
140  PRINT : INPUT "MAXIMUM JUNCTION TEMPERATURE? ";TJ
145  PRINT : PRINT "JUNCTION-CASE THERMAL RESISTANCE": INPUT "   (DEGREES
     /WATT)? ";JC
150  PRINT : PRINT "SELECT TYPE OF MOUNTING:"
155  PRINT "     1. DRY"
160  PRINT "     2. GREASE"
165  PRINT "     3. DRY WITH MICA WASHER"
170  INPUT "     4. GREASE WITH MICA WASHER ?";M
175  ON M GOTO 180,185,190,195
180  RC = 0.25:T$ = "DRY": GOTO 200
185  RC = 0.1:T$ = "GREASE": GOTO 200
190  RC = 0.8:T$ = "DRY WITH MICA WASHER": GOTO 200
195  RC = 0.4:T$ = "GREASE WITH MICA WASHER": GOTO 200
200  TS = TJ - P * (RC + JC)
205  RJ = (TJ - TA) / P
210  RS = RJ - (JC + RC)
215  TC = RC * P + TS:TJ = (RC + JC) * P + TS
220  IF TS <  = 55 THEN 240
225  TS = 55:RS = (TS - TA) / P:TC = RC * P + TS:TJ = (RC + JC) * P + TS
230  PRINT : INPUT "DO YOU WANT TO KNOW THE APPROXIMATE SINKVOLUME REQUIR
     ED (Y/N)? ";A$
235  IF A$ = "Y" THEN 250
240  GOSUB 275
245  GOTO 265
250  V = 46.155 * RS ^ ( - 1.52145):V =  INT (V * 100 + .5) / 100
255  GOSUB 275
260  PRINT : PRINT "APPROXIMATE HEAT SINK VOLUME": PRINT "REQUIRED = ";V;
     " CU IN"
265  PRINT : PRINT "REQUIRED HEAT SINK THERMAL": PRINT "RESISTANCE = ";RS
     ;" C/WATT OR LESS"
270  PRINT : END
275  CALL  - 936: PRINT  TAB( 16)"RESULTS"
280  PRINT : PRINT "USING...": PRINT T$: PRINT "CASE-SINK MOUNTING:"
285  PRINT : PRINT "MAXIMUM AMBIENT TEMPERATURE = ";TA;" C"
290  PRINT : PRINT "MAXIMUM JUNCTION TEMPERATURE = ";TJ;" C"
295  PRINT : PRINT "HEAT SINK TEMPERATURE = ";TS;" C"
300  PRINT : PRINT "CASE-SINK TEMPERATURE = ";TC;" C"
305  PRINT : PRINT "POWER DISSIPATION = ";P;" WATTS"
310  RETURN
```

Fig. 7-18. Listing for HEATSINK program.

3. Maximum ambient temperature.
4. Junction-case thermal resistance.

The maximum junction temperature and junction-case thermal resistance are usually given in the data sheet for the particular device that you are using. In addition, the program requires the method in which the device is to be mounted to the heatsink, which is one of the following:

1. Dry
2. Grease
3. Dry with mica washer
4. Grease with mica washer

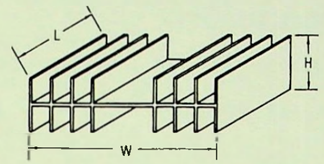


Fig. 7-19. Heat sink diagram.

The program requires all temperatures to be expressed in degrees Celsius, and determines the maximum thermal resistance of the required heat sink. As an alternative, the approximate volume of the heat sink may be calculated, using Fig. 7-19 as a guide for those who like to build their own, using aluminum.

Example 7-6

Determine the proper heat sink for a transistor that has the following electrical characteristics:

1. Dissipates 10 watts.
2. Maximum junction temperature = 135 degrees Celsius.
3. Junction-case resistance = 6.2 degrees/watt.

In addition, the maximum expected ambient temperature is 30 degrees, and the transistor is to be mounted with a mica washer with grease.

As shown in Fig. 7-20, the heat sink should have a thermal resistance of 2.5 degrees/watt or less, and which should have an approximate minimum volume of 11.45 in².

```
                HEAT SINK DESIGN
ALL TEMPERATURES ARE TO BE EXPRESSED IN
              DEGREES C

REQUIRED POWER DISSIPATION (WATTS)? 10
MAXIMUM AMBIENT TEMPERATURE? 30
MAXIMUM JUNCTION TEMPERATURE? 135
JUNCTION-CASE THERMAL RESISTANCE
   (DEGREES/WATT)? 6.2

SELECT TYPE OF MOUNTING:
     1. DRY
     2. GREASE
     3. DRY WITH MICA WASHER
     4. GREASE WITH MICA WASHER ?4

DO YOU WANT TO KNOW THE APPROXIMATE SINK
VOLUME REQUIRED (Y/N)? Y


                  RESULTS
USING...
GREASE WITH MICA WASHER
CASE-SINK MOUNTING:

MAXIMUM AMBIENT TEMPERATURE = 30 C

MAXIMUM JUNCTION TEMPERATURE = 121 C

HEAT SINK TEMPERATURE = 55 C

CASE-SINK TEMPERATURE = 59 C

POWER DISSIPATION = 10 WATTS

APPROXIMATE HEAT SINK VOLUME
REQUIRED = 11.45 CU IN

REQUIRED HEAT SINK THERMAL
RESISTANCE = 2.5 C/WATT OR LESS
```

Fig. 7-20. Results for Example 7-6.

8

Additional Routines

This final chapter presents several mathematical routines which aid the solution of a number of circuit problems. Included are the 4-quadrant arctangent function, mathematical operations on complex numbers, and the minimum and maximum value of an array.

4-QUADRANT ARCTANGENT FUNCTION

Although Apple II contains the arctangent function ATN (Z), it is only valid for angles between $-\pi/2$ (−90 degrees) and $+\pi/2$ (+90 degrees). As an example, if the argument Z is expressed as the ratio of Y and X coordinates, such that Z = Y/X, then ATN (Y/X) will yield correct results as long as X is greater than zero (i.e., first or fourth quadrant). If X is negative, ATN (Y/X) will still return a result that is between −1.57080 and +1.57080, which is in error, since the point actually lies in either the second or third quadrant.

The ARCTAN subroutine listed in Fig. 8-1 determines the 4-quadrant arctangent of the two variables X and Y and returns the answer expressed either in degrees or radians using the variables AD and AR respectively, as shown by the simple main program of Fig. 8-2. As will be

```
99   REM  SAMPLE MAIN PROGRAM
100  TEXT : HOME
101  INPUT "VALUE FOR X? ";X
102  INPUT "VALUE FOR Y? ";Y
103  GOSUB 6000
104  PRINT "THEN ATN(Y/X):"
105  PRINT "    ";AR;" RADIANS"
106  PRINT "    ";AD;" DEGREES"
108  PRINT : GOTO 101
109  END
```

Fig. 8-2. Simple main program for ARCTAN.

Fig. 8-1. Listing for ARCTAN subroutine.

```
6000  REM  4-QUADRANT ARCTANGENT SUBROUTINE (ARCTAN)
6001  REM  COPYRIGHT (C) 1980 HOWARD M. BERLIN, LTD
6002  REM  APPLE II....APPLESOFT VERSION
6003  REM   INPUTS:X AND Y
6004  REM   OUTPUTS:AR=ARCTAN IN RADIANS, AD=ARCTAN IN DEGREES
6005  IF Y <  > 0 THEN 6009
6006  IF X >  = 0 THEN 6008
6007  AR =  - 3.1415927: GOTO 6019
6008  AR = 0: GOTO 6019
6009  IF X <  = 0 THEN 6012
6010  IF Y = 0 THEN 6005
6011  AR =  ATN (Y / X): GOTO 6019
6012  IF X <  > 0 THEN 6016
6013  IF Y >  = 0 THEN 6015
6014  AR =  - 1.5707963: GOTO 6019
6015  AR = 1.5707963: GOTO 6019
6016  IF X >  = 0 THEN 6019
6017  IF Y = 0 THEN 6005
6018  AR = 3.1415927 +  ATN (Y / X)
6019  AD = AR * 57.2957795: RETURN
```

```
100  REM  COMPLEX ARITHMETIC PROGRAM (CPLXMATH)
101  HOME : VTAB (7): PRINT  TAB( 7): INVERSE : PRINT "COMPLEX ARITHMETIC
     PROGRAM": NORMAL
102  PRINT : PRINT  TAB( 11)"COPYRIGHT (C) 1980": PRINT : PRINT  TAB( 9)"
     HOWARD M. BERLIN, LTD."
103  PRINT : PRINT  TAB( 16)"APPLE II": PRINT : PRINT  TAB( 11)"APPLESOFT
     VERSION"
104  FOR II = 1 TO 2500: NEXT II
105  CLEAR
106  T = 57.2957795
107  HOME : PRINT "CHOOSE MATHEMATICAL OPERATION": PRINT "TO BE PERFORMED
     :"
108  PRINT : PRINT "   1. RECTANGULAR TO POLAR ": PRINT "      FOR A SING
     LE NUMBER."
109  PRINT : PRINT "   2. POLAR TO RECTANGULAR ": PRINT "      FOR A SING
     LE NUMBER."
110  PRINT : PRINT "   3. ADDITION OF TWO NUMBERS": PRINT "      (A + B)."
111  PRINT : PRINT "   4. SUBTRACTION OF TWO NUMBERS": PRINT "      (A -
     B)."
112  PRINT : PRINT "   5. MULTIPLICATION OF TWO NUMBERS": PRINT "      (A
      * B)."
113  PRINT : PRINT "   6. DIVISION OF TWO NUMBERS": PRINT "      (A / B)."
114  PRINT : INPUT "CHOICE? ";J: IF J < 3 THEN 127
115  HOME : PRINT : PRINT "SELECT INPUT FORMS OF THE 2 VARIABLES:"
116  PRINT : PRINT  TAB( 8)"A"; TAB( 22)"B"
117  PRINT : PRINT "1. RECTANGULAR   RECTANGULAR"
118  PRINT : PRINT "2. POLAR         POLAR"
119  PRINT : PRINT "3. POLAR         RECTANGULAR"
120  PRINT : PRINT "4. RECTANGULAR   POLAR"
121  PRINT : INPUT "CHOICE? ";I
122  IF J < 3 THEN 127
123  HOME : PRINT : PRINT "PRINT THE ANSWER IN:"
124  PRINT : PRINT "   1. RECTANGULAR FORM"
125  PRINT : PRINT "   2. POLAR FORM"
126  PRINT : PRINT "   3. BOTH FORMS ": PRINT : INPUT "CHOICE? ";K
127  HOME : PRINT : PRINT "ENTER:": GOSUB 182
128  IF J = 1 THEN I = 1
129  IF J = 2 THEN I = 2
130  ON I GOTO 131,132,133,134
131  AR = A1:AI = A2:BR = B1:BI = B2: GOTO 135
132  AM = A1:AA = A2:BM = B1:BA = B2: GOTO 135
133  AR = A1:AI = A2:BM = B1:BA = B2: GOTO 135
134  AM = A1:AA = A2:BR = B1:BI = B2
135  ON I GOTO 136,142,139,136
136  BM =  SQR (BR ^ 2 + BI ^ 2):Y = BI:X = BR: GOSUB 197
137  BA = ZD
138  IF I = 4 GOTO 143
139  AM =  SQR (AR ^ 2 + AI ^ 2):Y = AI:X = AR: GOSUB 197
140  AA = ZD
141  IF I <  > 3 GOTO 144
142  BR = BM *  COS (BA / T):BI = BM *  SIN (BA / T)
143  AR = AM *  COS (AA / T):AI = AM *  SIN (AA / T)
144  ON J GOTO 149,149,145,146,147,148
145  CR = AR + BR:CI = AI + BI: GOTO 149
146  CR = AR - BR:CI = AI - BI: GOTO 149
147  CM = AM * BM:CA = AA + BA: GOTO 149
148  CM = AM / BM:CA = AA - BA
149  ON J GOTO 155,155,150,150,153,153
```

Continued on next page.

Fig. 8-4. Listing for CPLXMATH program.


```
150  ON K GOTO 155,151,151
151  CM =  SQR (CR ^ 2 + CI ^ 2):X = CR:Y = CI: GOSUB 197
152  CA = ZD
153  ON K GOTO 154,158,154
154  CR = CM *  COS (CA / T):CI = CM *  SIN (CA / T)
155  AR =  INT (AR * 1000 + .5) / 1000:AI =  INT (AI * 1000 + .5) / 1000
156  CR =  INT (CR * 1000 + .5) / 1000:CI =  INT (CI * 1000 + .5) / 1000
157  AM =  INT (AM * 1000 + .5) / 1000:AA =  INT (AA * 10 + .5) / 10
158  CM =  INT (CM * 1000 + .5) / 1000:CA =  INT (CA * 10 + .5) / 10
159  ON J GOTO 160,162,164,170,172,179
160  PRINT : PRINT "THEN..."
161  PRINT : PRINT AR;" + J(";AI;") = ";AM;" / (";AA;")": GOTO 181
162  PRINT : PRINT "THEN..."
163  PRINT : PRINT AM;" / (";AA;") = ";AR;" + J(";AI;")": GOTO 181
164  C$ = "+": PRINT : PRINT "THEN...": PRINT : IF K - 2 >  = 0 THEN 166
165  PRINT "A ";C$;" B = ";CR;" +J(";CI;")": GOTO 181
166  IF K - 2 <  > 0 THEN 168
167  PRINT "A ";C$;" B = ";CM;" / (";CA;")": GOTO 181
168  PRINT "A ";C$;" B = ";CR;" + J(";CI;")"
169  PRINT "      = ";CM;" / (";CA;")": GOTO 181
170  C$ = "-": PRINT : PRINT "THEN...": PRINT : IF K - 2 < 0 THEN 165
171  GOTO 166
172  C$ = "*": PRINT : PRINT "THEN...": IF K - 2 < 0 THEN 174
173  GOTO 175
174  PRINT "A ";C$;" B = ";CR;" + J(";CI;")": GOTO 181
175  IF K - 2 <  > 0 THEN 177
176  PRINT "A ";C$;"B = ";CM;" / (";CA;")": GOTO 181
177  PRINT "A ";C$;" B = ";CR;" + J(";CI;")"
178  PRINT : PRINT "      = ";CM;" / (";CA;")": GOTO 181
179  C$ = "/": PRINT : PRINT "THEN...": PRINT : IF K - 2 < 0 THEN 174
180  GOTO 175
181  PRINT : PRINT : END
182  REM  INPUT SUBROUTINE
183  IF J = 1 THEN 187
184  IF J = 2 THEN 191
185  IF I = 1 OR I = 3 THEN 187
186  GOTO 191
187  PRINT : INPUT "REAL PART (A)? ";A1: PRINT : INPUT "IMAGINARY PART (A
     )? ";A2
188  IF J = 1 OR J = 2 THEN 190
189  GOTO 193
190  RETURN
191  PRINT : INPUT "MAGNITUDE (A)? ";A1: PRINT : INPUT "ANGLE (A)? ";A2
192  IF J = 1 OR J = 2 THEN  RETURN
193  IF I = 1 OR I = 4 THEN 195
194  GOTO 196
195  PRINT : INPUT "REAL PART (B)? ";B1: PRINT : INPUT "IMAGINARY PART (B
     )? ";B2: RETURN
196  PRINT : INPUT "MAGNITUDE (B)? ";B1: PRINT : INPUT "ANGLE (B)? ";B2: RETURN
197  REM   4-QUADRANT ARCTANGENT SUBROUTINE (ARCTAN)
198  IF Y <  > 0 THEN 202
199  IF X >  = 0 THEN 201
200  ZR =  - 3.1415927: GOTO 212
201  ZR = 0: GOTO 212
202  IF X <  = 0 THEN 205
203  IF Y = 0 THEN 198
204  ZR =  ATN (Y / X): GOTO 212
205  IF X <  > 0 THEN 209
206  IF Y >  = 0 THEN 207
```

Continued on next page.

Fig. 8-4 (cont). Listing for CPLXMATH program.

```
207  ZR =  - 1.5707963: GOTO 212
208  ZR = 1.5707963: GOTO 212
209  IF X >  = 0 THEN 210
210  IF Y = 0 THEN 198
211  ZR = 3.14159 +  ATN (Y / X)
212  ZD = ZR * 57.2957795: RETURN
```

Fig. 8-4 (cont). Listing for CPLXMATH program.

```
VALUE FOR X? 2
VALUE FOR Y? 5
THEN ATN(Y/X) :
   1.19028995 RADIANS
   68.1985905 DEGREES

VALUE FOR X? 2
VALUE FOR Y? -5
THEN ATN(Y/X):
   -1.19028995 RADIANS
   -68.1985905 DEGREES

VALUE FOR X? -2
VALUE FOR Y? 5
THEN ATN(Y/X):
   1.95130275 RADIANS
   111.801412 DEGREES

VALUE FOR X? -2
VALUE FOR Y? -5
THEN ATN(Y/X):
   4.33188265 RADIANS
   248.198593 DEGREES

VALUE FOR X?
```

Fig. 8-3. Results for Example 8-1.

discussed in the next section, the ARCTAN subroutine is necessary when converting rectangular coordinate, or complex numbers to their exponential or polar equivalents.

Example 8-1

Using the main program of Fig. 8-2, determine the arctangent in degrees and radians for the following X-Y coordinates:

X	Y
2	5
2	−5
−2	5
−2	−5

Fig. 8-3 gives the computer results.

COMPLEX NUMBER MATH

The ability to perform arithmetic operations with complex numbers in either rectangular or polar form can make any resulting computations easier. The CPLXMATH program shown in Fig. 8-4 will:

1. Convert a number from rectangular to polar form.
2. Convert a number from polar to rectangular form.
3. Add two complex numbers.
4. Subtract two complex numbers.
5. Multiply two complex numbers.
6. Divide one complex number by another.

By choosing either addition, subtraction, multiplication, or division, the program allows both numbers to be expressed in any combination of forms. In addition, the resulting answer can be chosen to be expressed in rectangular or polar form, or both.

```
CHOOSE MATHEMATICAL OPERATION
TO BE PERFORMED:

   1. RECTANGULAR TO POLAR
      FOR A SINGLE NUMBER.

   2. POLAR TO RECTANGULAR
      FOR A SINGLE NUMBER.

   3. ADDITION OF TWO NUMBERS
      (A + B).

   4. SUBTRACTION OF TWO NUMBERS
      (A - B).

   5. MULTIPLICATION OF TWO NUMBERS
      (A * B).

   6. DIVISION OF TWO NUMBERS
      (A / B).
CHOICE? 1

ENTER:
REAL PART (A)? 5
IMAGINARY PART (A)? -3
THEN...

5 + J(-3) = 5.831 / (-31)
```

Fig. 8-5. Results for Example 8-2.


```
CHOOSE MATHEMATICAL OPERATION
TO BE PERFORMED:

   1. RECTANGULAR TO POLAR
      FOR A SINGLE NUMBER.

   2. POLAR TO RECTANGULAR
      FOR A SINGLE NUMBER.

   3. ADDITION OF TWO NUMBERS
      (A + B).

   4. SUBTRACTION OF TWO NUMBERS
      (A - B).

   5. MULTIPLICATION OF TWO NUMBERS
      (A * B).

   6. DIVISION OF TWO NUMBERS
      (A / B).
CHOICE? 2

ENTER:
MAGNITUDE (A)? 6.92
ANGLE (A)? 56.1
THEN...

6.92/ (56.1) = 3.86 + J(5.744)
```

Fig. 8-6. Results for Example 8-3.

Example 8-2

Convert the complex number 5 − j3 to its equivalent polar form.

As shown in Fig. 8-5, the equivalent polar form has a magnitude of 5.831 and an angle of −31 degrees.

```
LENGTH OF THE ARRAY? 14
VALUE #1? 92
VALUE #2? 94
VALUE #3? 96
VALUE #4? 99
VALUE #5? 101
VALUE #6? 104
VALUE #7? 110
VALUE #8? 90
VALUE #9? 93
VALUE #10? 95
VALUE #11? 98
VALUE #12? 100
VALUE #13? 103
VALUE #14? 106

MINIMUM VALUE = 90
MAXIMUM VALUE = 110
```

Fig. 8-10. Results for Example 8-5.

```
CHOOSE MATHEMATICAL OPERATION
TO BE PERFORMED:

   1. RECTANGULAR TO POLAR
      FOR A SINGLE NUMBER.

   2. POLAR TO RECTANGULAR
      FOR A SINGLE NUMBER.

   3. ADDITION OF TWO NUMBERS
      (A + B).

   4. SUBTRACTION OF TWO NUMBERS
      (A - B).

   5. MULTIPLICATION OF TWO NUMBERS
      (A * B).

   6. DIVISION OF TWO NUMBERS
      (A / B).
CHOICE? 6

SELECT INPUT FORMS OF THE 2 VARIABLES:
        A            B
1. RECTANGULAR   RECTANGULAR
2. POLAR         POLAR
3. POLAR         RECTANGULAR
4. RECTANGULAR   POLAR
CHOICE? 3

PRINT THE ANSWER IN:
   1. RECTANGULAR FORM
   2. POLAR FORM
   3. BOTH FORMS
CHOICE? 3

ENTER:
REAL PART (A)? 5
IMAGINARY PART (A)? -3
MAGNITUDE (B)? 6.92
ANGLE (B)? 56.1
THEN...

A / B = .043 + J(-.842)

      = .843 / (-87.1)
```

Fig. 8-7. Results for Example 8-4.

```
100  REM  SAMPLE MAIN PROGRAM
101  DIM X(100)
102  HOME : INPUT "LENGTH OF ARRAY? ";N
103  FOR I = 1 TO N
104  PRINT "VALUE #";I;"? ";
105  INPUT "";X(I)
106  NEXT I
107  GOSUB 7000
108  PRINT : PRINT "MINIMUM VALUE = ";X1
109  PRINT "MAXIMUM VALUE = ";X2
110  END
```

Fig. 8-9. Simple main program for MINMAX.

```
7000  REM   MINIMUM AND MAXIMUM VALUES OF AN ARRAY (MINMAX)
7001  REM   COPYRIGHT (C) 1980,HOWARD M. BERLIN, LTD
7002  REM   APPLE II,APPLESOFT VERSION
7003  REM   X(I)=INPUT POINTS
7004  REM   N=LENGTH OF THE ARRAY
7005  REM   X1=MINIMUM VALUE  X2=MAXIMUM VALUE
7006  X1 = X(1):X2 = X(1)
7007  FOR I = 2 TO N
7008  IF (X1 - X(I)) <  = 0 THEN 7010
7009  X1 = X(I): GOTO 7012
7010  IF (X2 - X(I)) >  = 0 THEN 7012
7011  X2 = X(I)
7012  NEXT I
7013  RETURN
```

Fig. 8-8. Listing for MINMAX subroutine.

Example 8-3

Convert to rectangular form the polar number that has a magnitude of 6.92 and an angle of 56.1 degrees.

As shown in Fig. 8-6, the equivalent complex number in rectangular form is 3.86 + j5.744.

Example 8-4

Divide the number "A", 5 − j3, by the number "B," having a magnitude of 6.92 and an angle of 56.1 degrees, and express the answer in both rectangular and polar form.

As shown in Fig. 8-7, we enter the number "3" when asked to select the forms of the two numbers, as A and B are in rectangular and polar forms respectively.

MINIMUM AND MAXIMUM VALUES OF AN ARRAY

The *MINMAX* subroutine of Fig. 8-8 determines the minimum and maximum values of the array *X(I)*. This routine is used in many of the plotting routines of Chapter 2, the DAMPED program of Chapter 3, and the DSTATS program of Chapter 4. The required input variables are the array and its length N. When control returns to the main program, the minimum and maximum values are found in the variables X1 and X2 respectively.

Example 8-5

Using the simple main program of Fig. 8-9, determine the minimum and maximum values of the values listed in Table 2-1. The resulting output is shown in Fig. 8-10.

APPENDIX

Standard Resistor and Capacitor Values

RESISTORS

1. The following ±5% standard decade values are available. Those marked with * are the ones most readily available from electronic suppliers.

1.0*	1.8*	3.3*	5.6*
1.1	2.0	3.6	6.2
1.2*	2.2*	3.9*	6.8*
1.3	2.4	4.3	7.5
1.5*	2.7*	4.7*	8.2*
1.6	3.0	5.1	9.1

To obtain standard resistance values, multiply preferred number from decade table by powers of 10. Standard values are available from 10 Ω to 22 MΩ.

2. The following ±1% values are available, but at a higher cost.

10.0	12.1	14.7	17.8	21.5	26.1	31.6	38.3	46.4	56.2	68.1	82.5
10.2	12.4	15.0	18.2	22.1	26.7	32.4	39.2	47.5	57.6	69.8	84.5
10.5	12.7	15.4	18.7	22.6	27.4	33.2	40.2	48.7	59.0	71.5	86.6
10.7	13.0	15.8	19.1	23.2	28.0	34.0	41.2	49.9	60.4	73.2	88.7
11.0	13.3	16.2	19.6	23.7	28.7	34.8	42.2	51.1	61.9	75.0	90.0
11.3	13.7	16.5	20.0	24.3	29.4	35.7	43.2	52.3	63.4	76.8	93.1
11.5	14.0	16.9	20.5	24.9	30.1	36.5	44.2	53.6	64.9	78.7	95.3
11.8	14.3	17.4	21.0	25.5	30.9	37.4	45.3	54.9	66.5	80.6	97.6

These standard values are available from 10 Ω to 22.1 MΩ.

CAPACITORS

In general, capacitor values follow the standard decade values for ±10% resistors. The values listed below are in microfarads, and those marked with * are the ones most readily available from electronic suppliers.

.001*	.0033*	.01*	.033*	.1*	.33*
.0012	.0039	.012	.039	.12	.39
.0015	.0047*	.015	.047*	.15	.47*
.0018	.005	.018	.05	.18	.5
.002	.0056	.02	.056	.2	.56
.0022*	.0068*	.022*	.068*	.22*	.68*
.0025	.0075	.025	.075	.25	.75
.0027	.0082	.027	.082	.27	.82

Index

A

Applesoft, 7
ARCTAN
  program, 123
  subroutine, 123
Arctangent function, 4-quadrant, 123
Array
  maximum values, 128
  minimum values, 128
ATTNPAD program, 68

B

Bandpass filter(s), 97-99
BIAS program, 116
BNDPASS1 program, 98
BNDPASS 2 program, 101
Butterworth bandpass filters, 99

C

Cartesian plots, 11-16, 25
CPLXMATH program, 124
CPLXSEQ program, 80
Current network analysis, mesh, 72
Curves, least squares regression, 47-56

D

Daisywheel printer, 8
Damped
  oscillations, analysis, 46
  program, 44
Delta-Wye transforms, 72
DFTRANS program, 41
DIM statement, 16
Disk operating system, 7
Diskette, 7
DOS, 7
Dot-matrix printer, 8
DSTATS program, 58

F

Filter(s)
  bandpass, 96-99
  Butterworth, 99
  high pass, 83
  low pass, 83
  notch, 109
  state variable, 83
  wideband, 109
555 timer design, 112
4-quadrant arctangent function, 123
Fourier
  series, 38-40
  transform, 43-46
FSERIES program, 37

G

Graphics, 7
  HiRes, 7
  LoRes, 7

H

Heat sink design, 119-121
HEATSINK program, 120
HGR, 7
High-pass filters, 83
HiRes graphics, 7
Histograms, 9, 11

I

ILAPLACE program, 65
Impedance matching pads, 67-72
Inverse laplace transform, 61-67

L

Laplace transform, inverse, 61-67
LCPAD program, 71
LCPSPAD program, 73
Line
  printer, 7
    plotting, 22-26
LINEPLOT program, 24
Log-log plots, 18
LOGLOG subroutine, 19
LoRes graphics, 7
Lossless pads, 70-72
Low-pass filters, 83
LPHP program, 84

M

Matching pads, impedance, 67-72
Math, complex number, 126-128
Mesh current network analysis, 72
MINMAX program, 127
MLREGRESS program, 55

N

Network analysis, mesh current, 72
Notch
  filter, 109
  program, 107

O

Oscillations
  analysis, 46
  damped, 46

P

Pads
  impedance matching, 67-72
  lossless, 70-72
  resistance, 67-70

Parallel line printer, 9
PCARTXY subroutine, 12
PHISTGM
  program, 11
  subroutine, 10
Pi-Tee transforms, 72
Plots
  Cartesian, 11-16, 25
  log-log, 18
  polar, 18-20, 26
  semilogarithmic, 16-18, 25-26
Plotting with line printer, 22-26
PLYROOTS program, 62
Polar plots, 18-20, 26
POLARPLT program, 30
Polynomials, roots, 61
Printer
  daisywheel, 8
  dot-matrix, 8
  line, 7
  parallel line, 9
  plotting with, 22-26
  serial line, 9
Program
  ARCTAN, 123
  ATTNPAD, 68
  BIAS, 116
  BNDPASS1, 98
  BNDPASS2, 101
  CPLXMATH, 124
  CPLXSEQ, 80
  DAMPED, 44
  DFTRANS, 41
  DSTATS, 58
  FSERIES, 37
  HEATSINK, 120
  ILAPLACE, 65
  LCPAD, 71
  LCPSPAD, 73
  LINEPLOT, 24
  LPHP, 84
  MINMAX, 127
  MLREGRESS, 55
  NOTCH, 107
  PHISTGM, 11
  PLYROOTS, 62
  POLARPLT, 30
  PSEMILOG, 16
  REGRESSN, 49
  RMSAV, 34
  SIMULEQU, 77
  SVFILTER, 94
  TEEPI, 75
  TIMER555, 113
  TRANSAMP, 117
  TUNEDBP, 105
  VPOLAR, 21

Program—cont
  XLOGPLOT, 27
  ZENER, 111
PSEMILOG
  program, 16
  subroutine, 15

R

Regression
  curves, least squares, 47-56
  program, 49
Regulator design, zener diode voltage, 111-112
Resistance pads, 67-70
RMSAV program, 34
Roots, polynomials, 61

S

Semilogarithmic plots, 16-18, 25-26
Serial line printer, 9
SIMULEQU program, 77
Staggered-tuned Butterworth filters, 99
State-variable filter, 83
Statistics, sample, 56
Subroutines, 9
SVFILTER program, 94

T

TEEPI program, 75
Timer design, 555, 112
TIMER555 program, 113
TRANSAMP program, 117
Transform(s)
  Delta-Wye, 72
  Fourier, 43-46
  inverse Laplace, 61-67
  Pi-Tee, 72
Transistors, 114-119
TUNEDBP program, 105

V

Variables, relationship, 47-56
Voltage regulator design, zener diode, 111-112
VPOLAR program, 21

W

Waveform
  average, 33-38
  rms, 33-38
Wideband filter, 109

X

XLOGPLOT program, 27

Z

Zener
  diode voltage regulator design, 111-112
  program, 111

CIRCUIT DESIGN SOFTWARE

Engineers, designers, and serious hobbyists running circuit design data on an Apple II Plus or an Apple II with Applesoft can get a lot of use from 37 disk-based circuit design programs that save time on routine calculations, statistics, and analyses.

Each program has been thoroughly tested and debugged by author/engineer/educator Howard Berlin, and appears in one of 7 different packages produced by Advanced Operating Systems, computer software division of the Howard W. Sams Company.

Programs can be used by themselves or as subroutines in a larger program. At least 32K RAM is required. The programs and packages are as follows:

ELECTRONICS I package

A total of 5 programs covering zener diode voltage-regulator design, 555 timer design (monostable and astable circuits), transistor bias parameters, single-state transistor amplifier design, and heat-sink selection and design. Ask for No. 26012.

ELECTRONICS II package

Seven programs in all, dealing with four-quadrant arctangent function, rectangular/polar conversion and complex number mathematics, minimum and maximum values of an array, roots of polynomials with real coefficients, Inverse Laplace transforms of a transfer function, and solution of simultaneous equations with real and complex coefficients. Ask for No. 26013.

ELECTRONICS III package

Provides average and rms values of a periodic function, Fourier series expansion of a periodic function, Fourier transform, analysis of damped oscillations, attenuator pad design, lossless L-C pad design, lossless L-C pad with phase shift, and delta-wye transforms. Total of 8 programs. Ask for No. 26014.

PLOTTING GRAPHS FOR LINE PRINTER package

Three programs that allow plotting of cartesian plots, semilogarithmic plots, and polar plots by line printers. Ask for No. 26009.

ACTIVE FILTER DESIGN package

Six programs dealing with low- and high-pass filters (Bessel, Butterworth, 1-, 2-, and 3-dB Chebyshev), state-variable filter, bandpass filter with Q's less than 10 and 50, staggered-tuned Butterworth bandpass filter (2, 3, or 4 stages), and notch filter. Ask for No. 26010.

DESCRIPTIVE STATISTICS & REGRESSION ANALYSIS package

Includes descriptive statistics (mean, standard deviation, variance, kurtosis, Z-scores), curvilinear regression (linear, inverse, nth order polynomial, exponential, geometric), and multiple linear regression in a series of 3 programs. Ask for No. 26011.

PLOTTING GRAPHS FOR VIDEO DISPLAY package

A series of 5 programs that display histograms, cartesian plots, semilogarithmic plots, log-log plots, and polar plots on a video monitor. Ask for No. 26015.

READER SERVICE CARD

To better serve you, the reader, please take a moment to fill out this card, or a copy of it, for us. Not only will you be kept up to date on the Blacksburg Series books, but as an extra bonus, **we will randomly select five cards every month, from all of the cards sent to us during the previous month. The names that are drawn will win, absolutely free, a book from the Blacksburg Continuing Education Series.** Therefore, make sure to indicate your choice in the space provided below. For a complete listing of all the books to choose from, refer to the inside front cover of this book. Please, one card per person. Give everyone a chance.

In order to find out who has won a book in your area, call (703) 953-1861 anytime during the night or weekend. When you do call, an answering machine will let you know the monthly winners. Too good to be true? Just give us a call. Good luck.

If I win, please send me a copy of:

_____

I understand that this book will be sent to me absolutely free, if my card is selected.

For our information, how about telling us a little about yourself. We are interested in your occupation, how and where you normally purchase books and the books that you would like to see in the Blacksburg Series. We are also interested in finding authors for the series, so if you have a book idea, write to The Blacksburg Group, Inc., P.O. Box 242, Blacksburg, VA 24060 and ask for an Author Packet. We are also interested in TRS-80, APPLE, OSI and PET BASIC programs.

My occupation is _____
I buy books through/from _____
Would you buy books through the mail? _____
I'd like to see a book about _____
Name _____
Address _____
City _____
State _____ Zip _____

MAIL TO: BOOKS, BOX 715, BLACKSBURG, VA 24060
!!!!!PLEASE PRINT!!!!!

21863

The Blacksburg Group

According to Business Week magazine (Technology July 6, 1976) large scale integrated circuits or LSI "chips" are creating a second industrial revolution that will quickly involve us all. The speed of the developments in this area is breathtaking and it becomes more and more difficult to keep up with the rapid advances that are being made. It is also becoming difficult for newcomers to "get on board."

It has been our objective, as The Blacksburg Group, to develop timely and effective educational materials that will permit students, engineers, scientists, technicians and others to quickly learn how to use new technologies and electronic techniques. We continue to do this through several means, textbooks, short courses, seminars and through the development of special electronic devices and training aids.

Our group members make their home in Blacksburg, found in the Appalachian Mountains of southwestern Virginia. While we didn't actively start our group collaboration until the Spring of 1974, members of our group have been involved in digital electronics, minicomputers and microcomputers for some time.

Some of our past experiences and on-going efforts include the following:

–The design and development of what is considered to be the first popular hobbyist computer. The Mark-8 was featured in Radio-Electronics magazine in 1974. We have also designed several 8080-based computers, including the MMD-1 system. Our most recent computer is an 8085-based computer for educational use, and for use in small controllers.

–The Blacksburg Continuing Education Series™ covers subjects ranging from basic electronics through microcomputers, operational amplifiers, and active filters. Test experiments and examples have been provided in each book. We are strong believers in the use of detailed experiments and examples to reinforce basic concepts. This series originally started as our Bugbook series and many titles are now being translated into Chinese, Japanese, German and Italian.

–We have pioneered the use of small, self-contained computers in hands-on courses for microcomputer users. Many of our designs have evolved into commercial products that are marketed by E&L Instruments and PACCOM, and are available from Group Technology, Ltd., Check, VA 24072.

–Our short courses and seminar programs have been presented throughout the world. Programs are offered by The Blacksburg Group, and by the Virginia Polytechnic Institute Extension Division. Each series of courses provides hands-on experience with real computers and electronic devices. Courses and seminars are provided on a regular basis, and are also provided for groups, companies and schools at a site of their choosing. We are strong believers in practical laboratory exercises, so much time is spent working with electronic equipment, computers and circuits.

Additional information may be obtained from Dr. Chris Titus, the Blacksburg Group, Inc. (703) 951-9030 or from Dr. Linda Leffel, Virginia Tech Continuing Education Center (703) 961-5241.

Our group members are Mr. David G. Larsen, who is on the faculty of the Department of Chemistry at Virginia Tech, and Drs. Jon Titus and Chris Titus who work full-time with The Blacksburg Group, all of Blacksburg, VA.

S S0 0 6 1 4 3 4 4 7

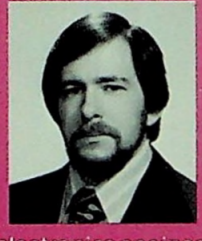
CIRCUIT DESIGN PROGRAMS FOR THE APPLE II®

BY
HOWARD M. BERLIN

A variety of useful BASIC language programs that will greatly simplify the design and analysis of commonly encountered problems related to plotting and the simple statistical verification of experimental data are presented in this book.

Featured programs in this book permit:

- The design of filters.
- The analysis of 2-port networks.
- The inverse Laplace transform.
- Real and imaginary roots of polynomials.
- Pi-Tee (Delta-Wye) transformations.
- The solution of mesh and node equations.



HOWARD M. BERLIN is an electronics engineer with the U.S. Army Material Systems Analysis Activity at Aberdeen Proving Ground, Maryland, primarily involved in biomedical engineering research and physiological instrumentation. He has taught short courses at several universities and conferences, and is the author of a number of governmental reports and articles in both scientific journals and amateur radio magazines. He is the author of the following Sams books: The 555 Timer Applications Sourcebook, With Experiments; Design of Active Filters, With Experiments; Design of Op-Amp Circuits, With Experiments; Design of Phase-Locked Loop Circuits, With Experiments; Guide to CMOS Basics, Circuits, & Experiments; Circuit Design Programs for the TRS-80®; and coauthor with Robert T. Stone of Design of VMOS Circuits, With Experiments. He is a member of Sigma Xi, the Delaware Academy of Medicine, New York Academy of Science, and a Senior Member of IEEE. In his spare time, Howard enjoys playing bluegrass banjo and folk guitar, and is an active amateur radio operator (W3HB).

Howard W. Sams & Co., Inc.
4300 W. 62nd Street, Indianapolis, Indiana 46268 USA
$15.95/21863 ISBN: 0-672-21863-1